



# Beyond core count: a look at new mainstream computing platforms for HEP workloads

> 15/10/2013

CHEP 2013

P. Szostek, A. Nowak, G. Bitzes,  
L. Valsan, S. Jarp,



**CERN**openlab

- Hardware and software status in mainstream HEP computing
- Performed tests
  - Hardware
  - Selected Benchmarks
- Results
  - A look at the Intel „Haswell” micro-architecture
  - Future trends
  - Conclusions

# HW&SW status in mainstream HEP computing

- Top of the class
  - dual socket nodes with 24 cores
  - quad socket (AMD) with 64 cores.
  - (CHEP2010: 8 cores)
- Commonly used: dual socket, 2x8 cores
- Difficulties in harnessing micro-parallelism
  - Still very little multi-threaded software (see: HEPSPEC06)
    - › Each core used for a separate process, no communication between processes, huge memory footprint
  - Hyperthreading (SMT) challenges
    - › In many parallel applications SMT means 10-30% more performance for free! But double memory cost in a multi-process model.
  - (auto-)vectorization

# Hardware setup for tests

## Intel server CPUs

- “Sandy Bridge” E5-2690
- “Ivy Bridge” E5-2695 v2
- 2 sockets: 16 and 24 cores
- Shrink from 32 to 22nm
- Same cache, lower TDP

## Intel workstation CPUs

- “Ivy Bridge” E3-1265L v2
- “Haswell” E3-1285L v3
- Single socket: 4 cores, 8 threads
- New micro-architecture
  - AVX2
  - Wider core (4<sup>th</sup> ALU, 3<sup>rd</sup> AGU, 2<sup>nd</sup> branch prediction unit)

# Used benchmarks

- Power consumption
- HEPSPEC06
  - performance per watt and scalability
  - HEP-world standard
- Multi-threaded Geant4 prototype
  - scalability
  - toolkit for simulating the passage of particles through matter
- Maximum Likelihood Fit
  - scalability
- Alice/CBM Trackfitter

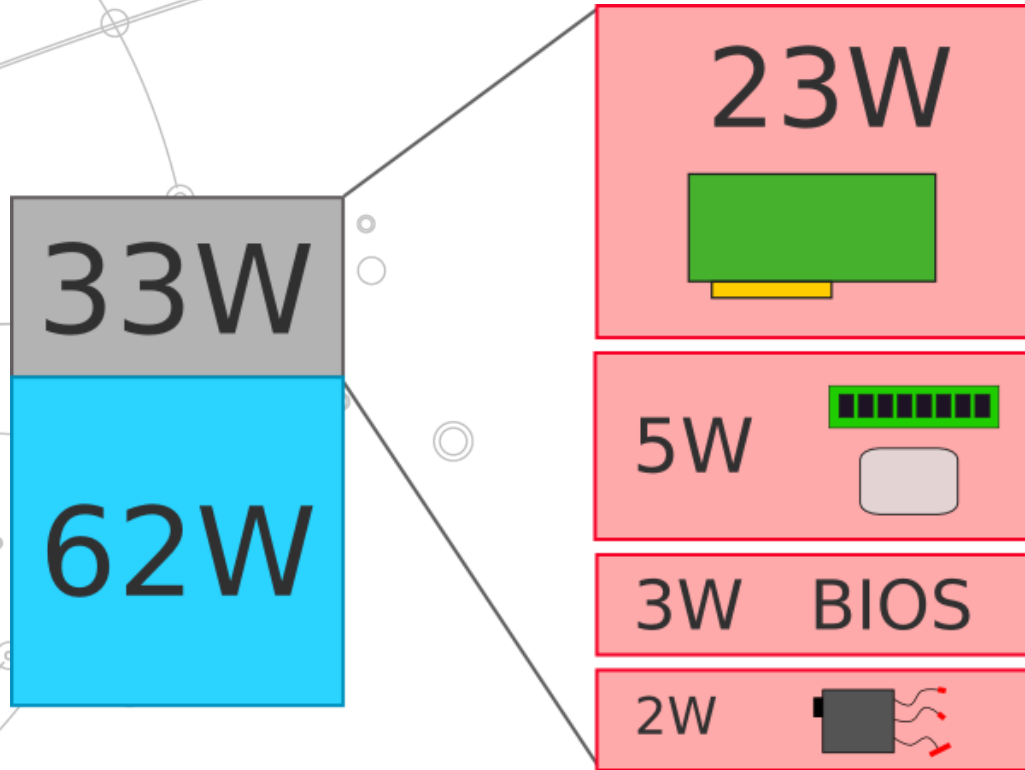
# Power consumption measurements

SKU	Architecture	Turbo on	Turbo on	Turbo off	Turbo off
		SMT on	SMT off	SMT on	SMT off
E3-1265L V2	Ivy Bridge	62	60	60	60
E3-1285L V3	Haswell	63	63	63	60
E5-2690	Sandy Bridge	433	421	421	375
E5-2595 V2	Ivy Bridge	403	406	338	321

Power consumption in watts under full load

- Workstations: power consumption at the same level
- Servers: 16 threads more and significant power consumption decrease!

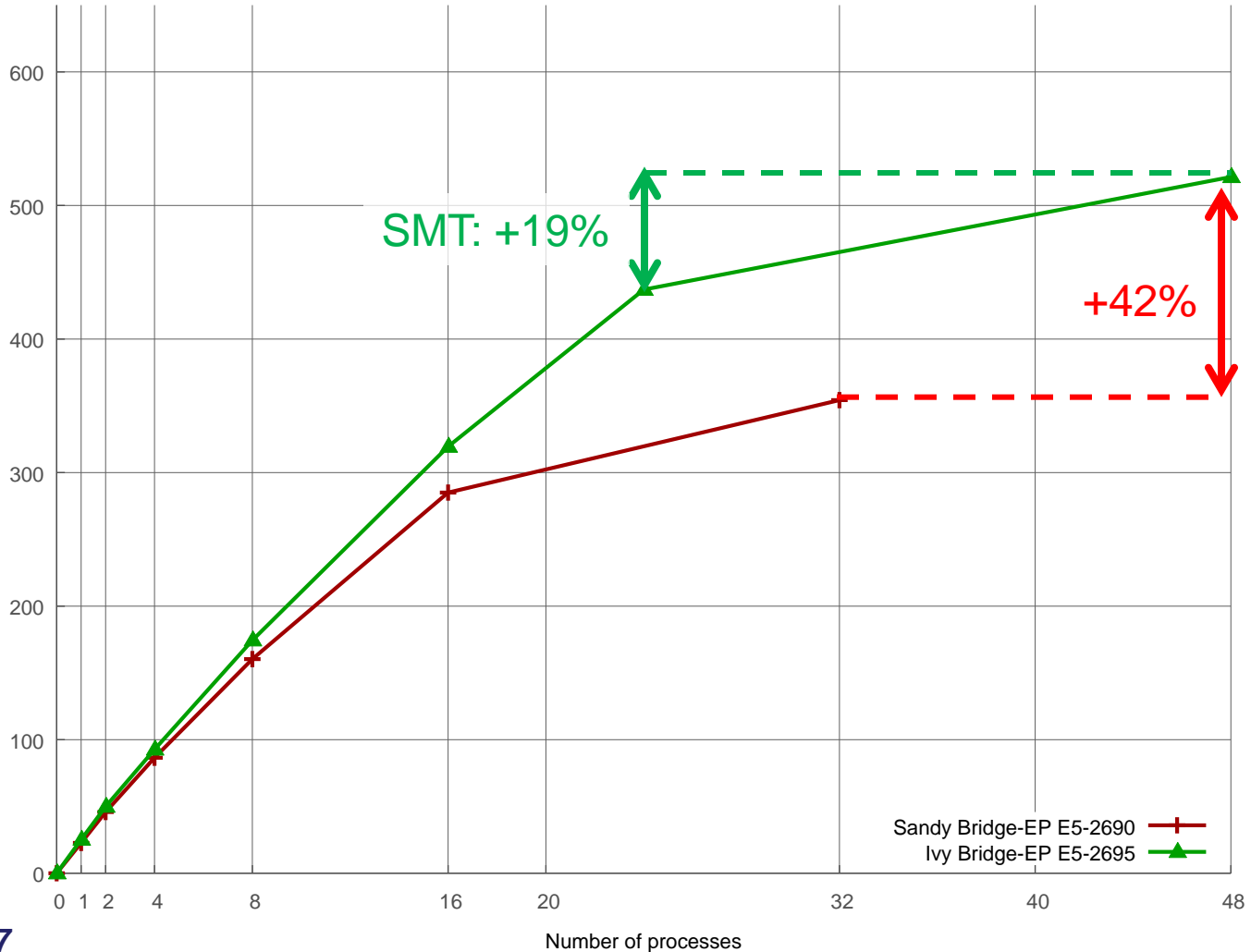
# Power consumption – lesson learnt



Optimizing components for comparison purposes:  
GPU (24%), RAM (5%), BIOS update (3%), PSU (2%)

(% of power consumption)

# HEPSPEC06 – „Sandy Bridge” vs. „Ivy Bridge” (frequency scaled)



gcc 4.4.7  
-fPIC -O2

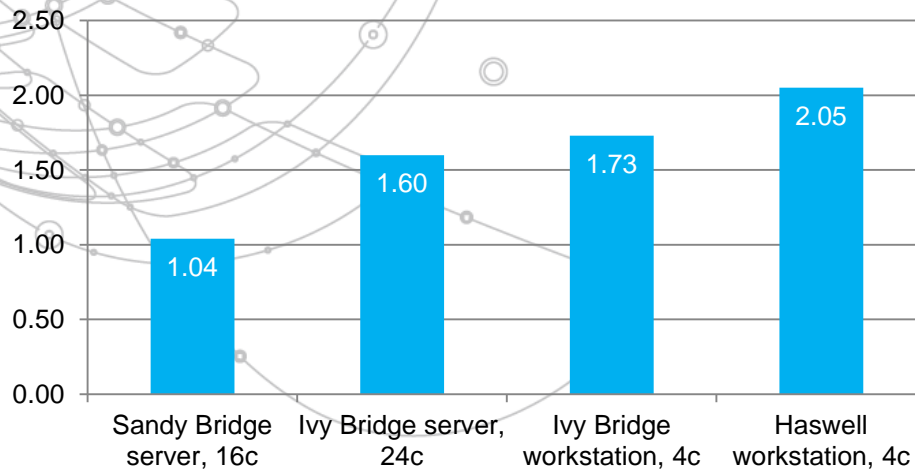


- Scaled results for workstations almost identical
  - Waiting for first “Haswell” servers
- Results with gcc 4.8.1 ~1% better
- Is HEPSPEC06 still a good benchmark?
  - If multi-threading pays off much better than multi-processing, should we use single threaded test workloads?
  - Will there be a revolution in memory technology that will delay the drive towards parallelism?

# HEPSPEC06 per Watt

	„Sandy Bridge” server E5-2690	„Ivy Bridge” server E5-2690 V2	„Ivy Bridge” workstation (E3-1265L V2)	„Haswell” workstation E3-1285L V3
HS06	381	463	94	115
Standard energy measurement	362	290	54	56
HS06 per Watt	1.04	1.60	1.73	2.06

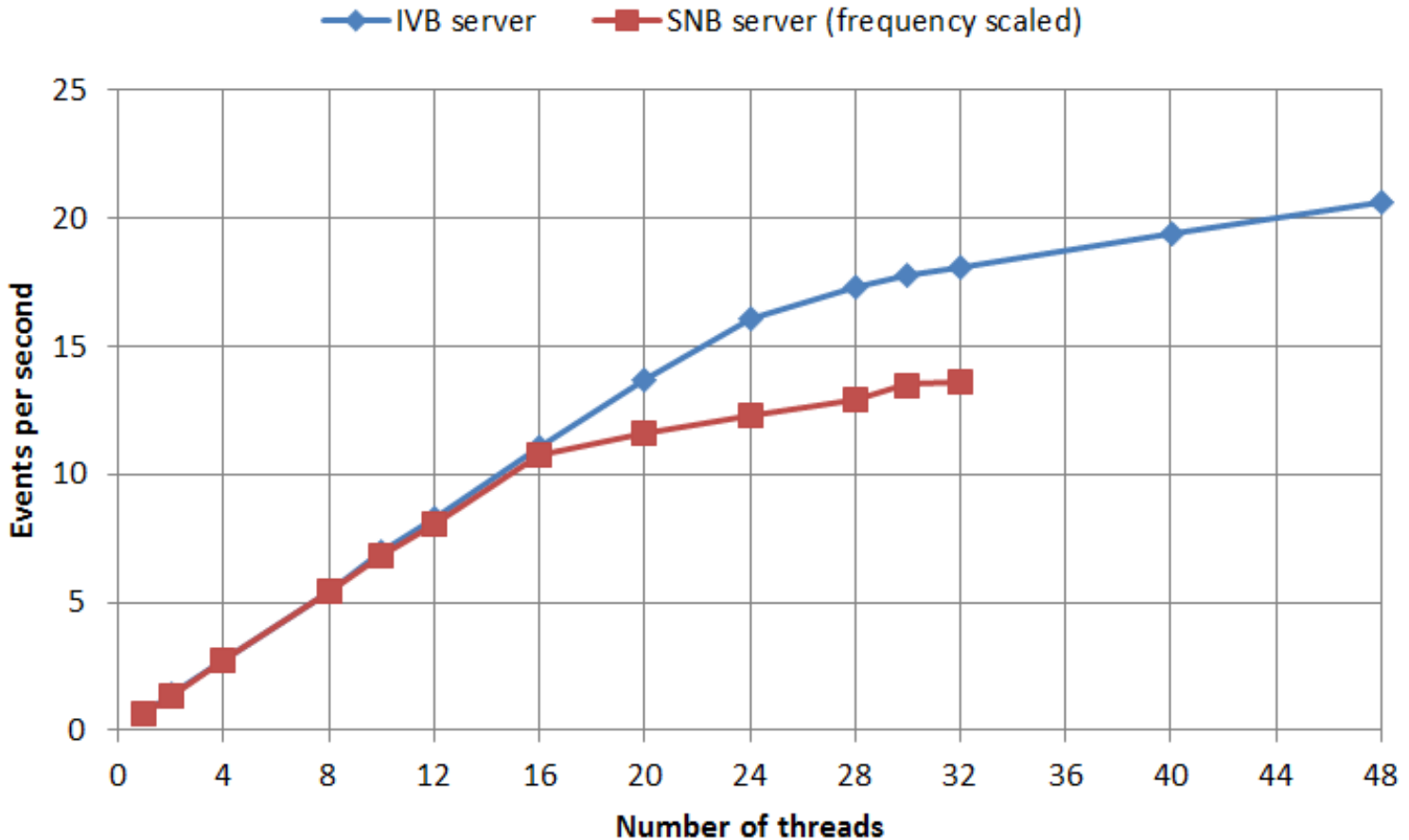
## HEPSPEC06/W (higher is better)



- From IVB to HSW: 20% improvement (incl. motherboard)
- From SNB-EP to IVB-EP: **54%**
- High values for desktops are due to manually optimized energy use: barebone systems

# New multi-threaded Geant 4 prototype

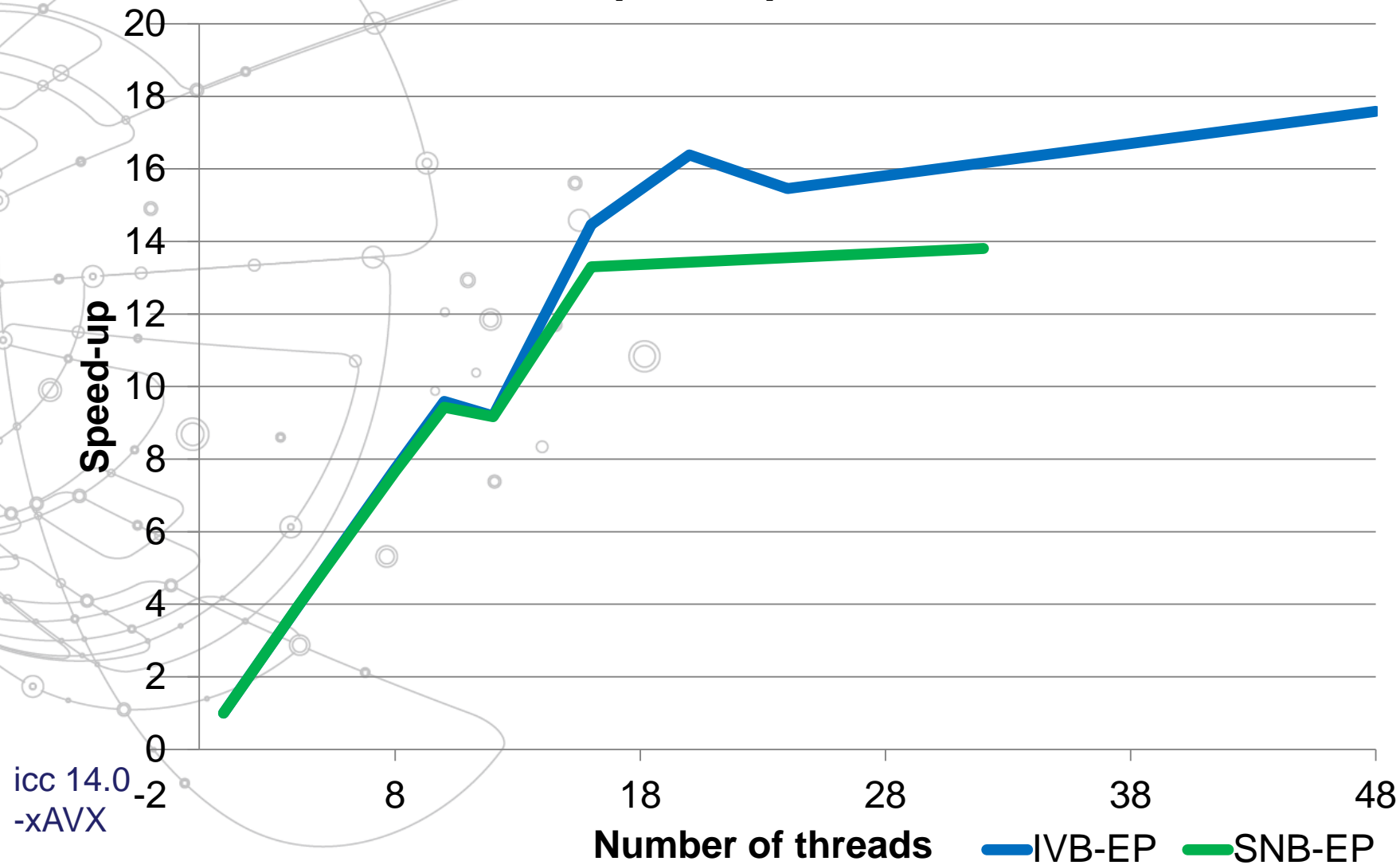
New MTG4 prototype - pi@50GeV events/s  
("weak scaling" - preliminary results)



icc 13.0.1

-O2 -fp-model-precise -ansi -fPIC

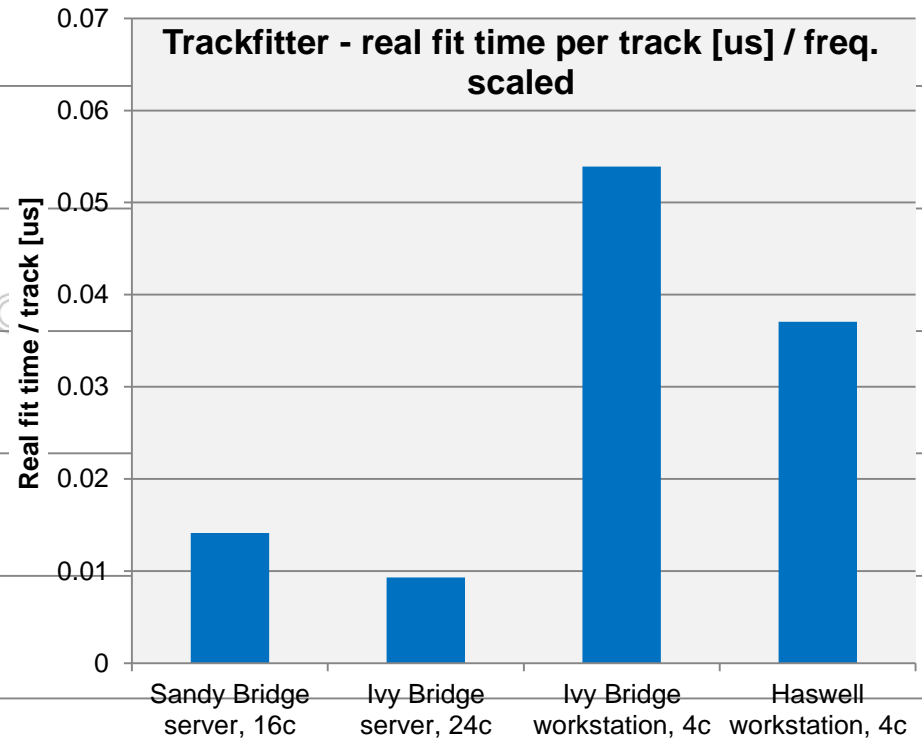
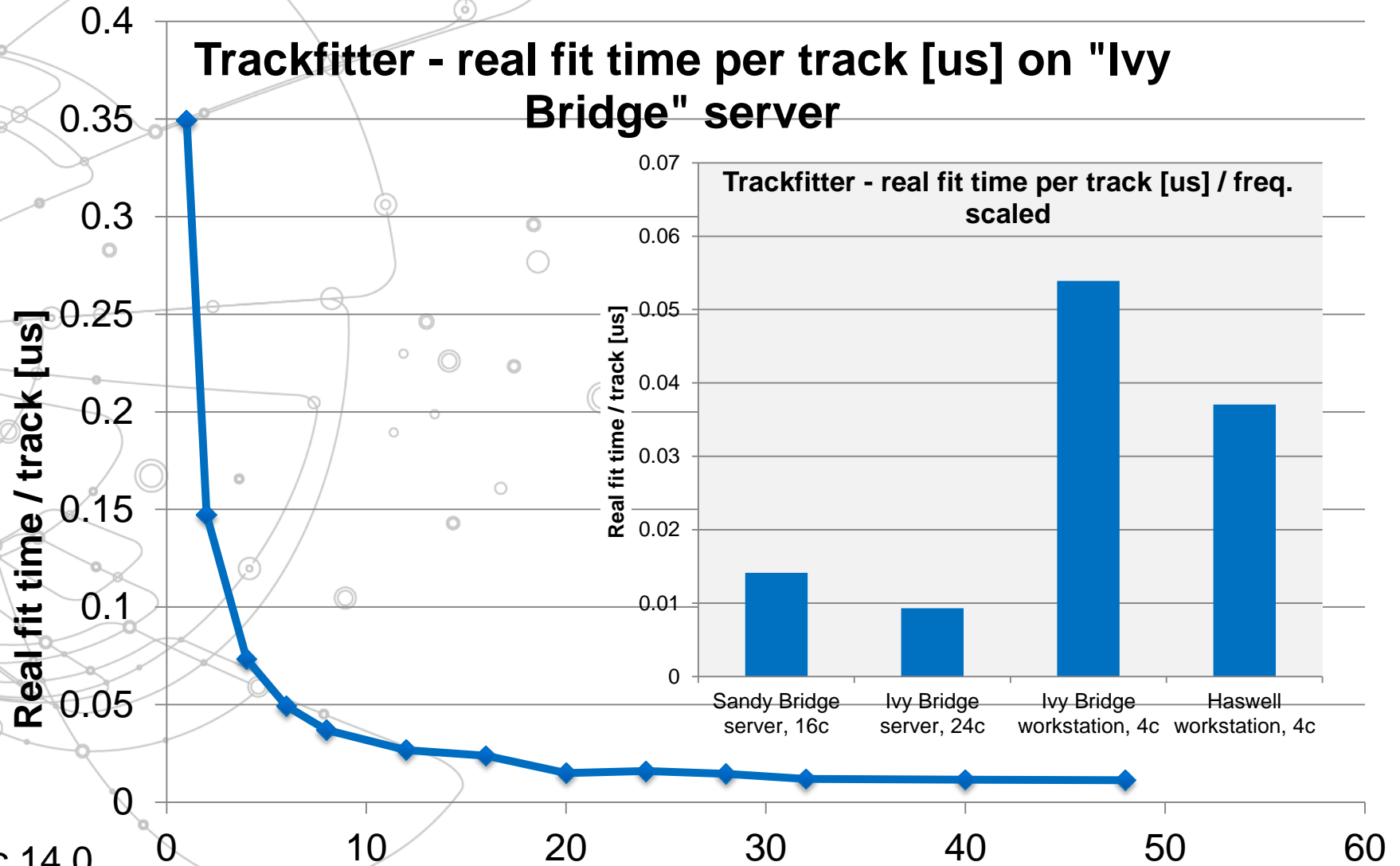
## MLFit speed-up with AVX



icc 14.0  
-xAVX

# ALICE/CBM Trackfitter

## Trackfitter - real fit time per track [us] on "Ivy Bridge" server



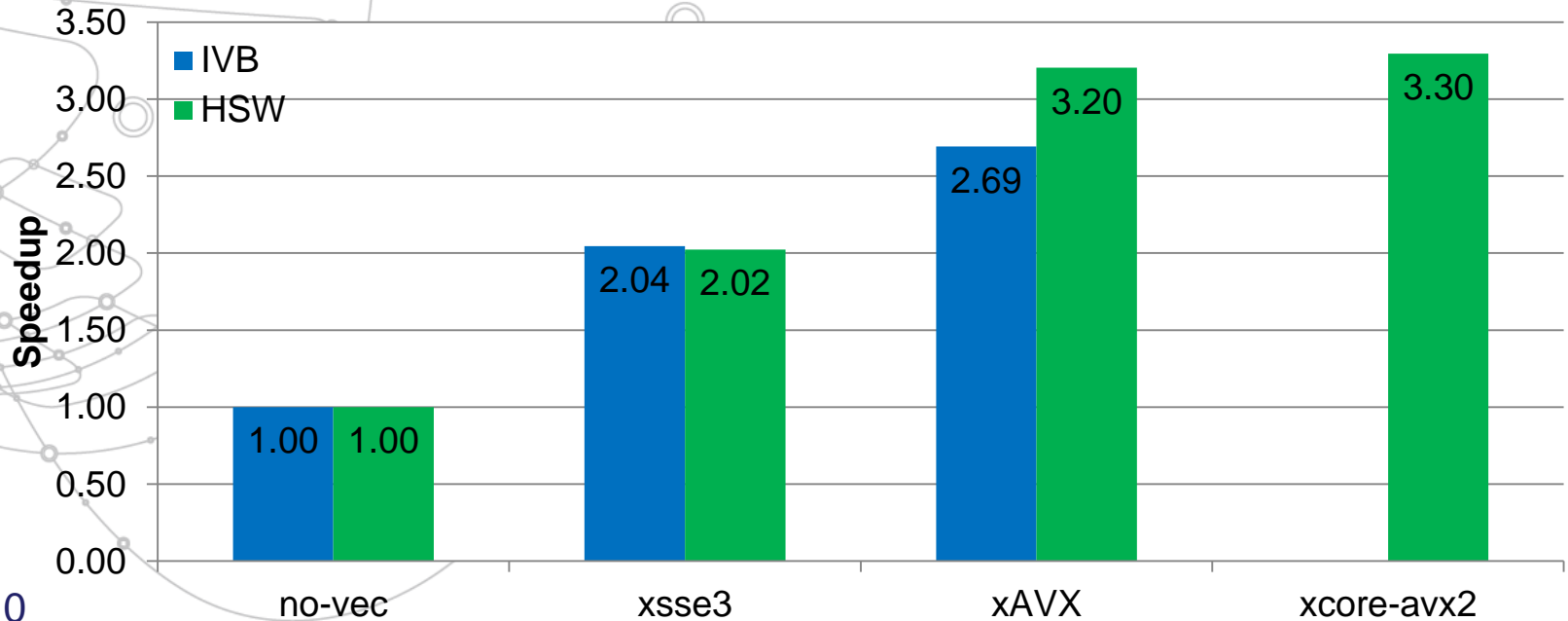
lcc 14.0  
-O3 -inline-forceinline -core-avx2

# SIMD from the Haswell perspective

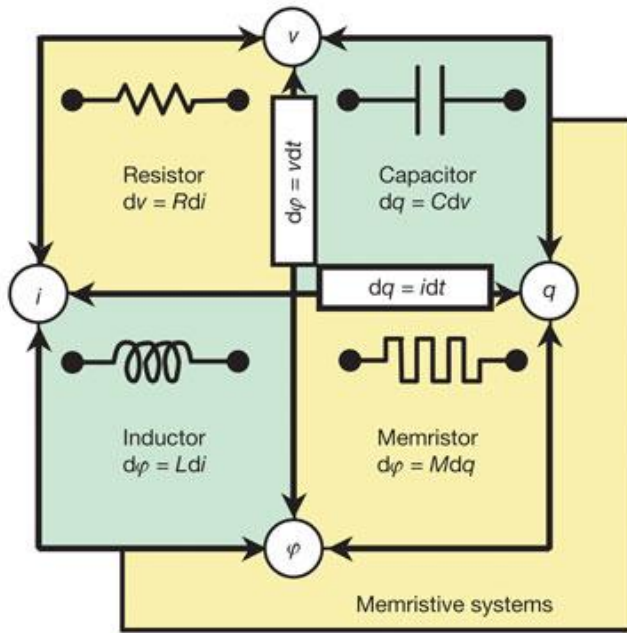
- FMA now fully supported + extra execution ports

MLFit DP vectorization speedup on "Ivy Bridge" server and "Haswell" desktop

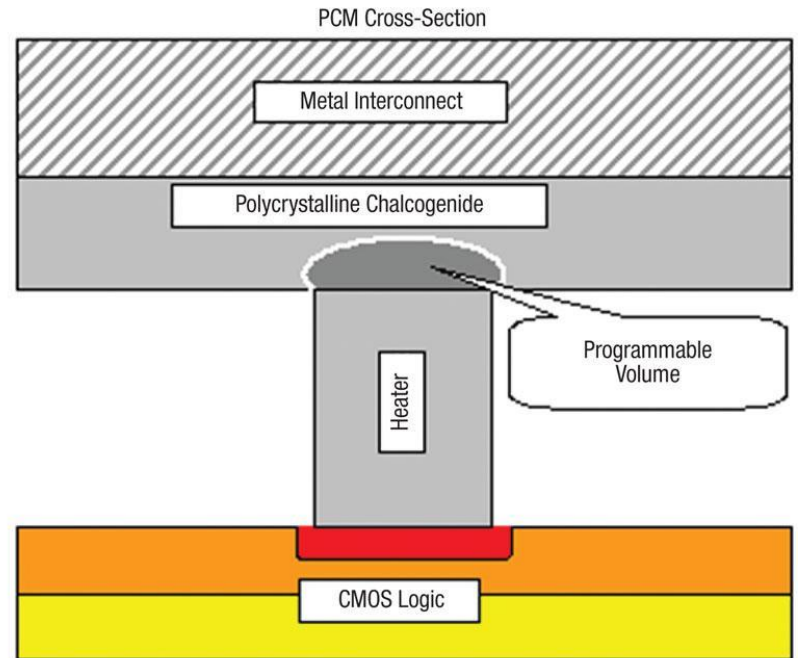
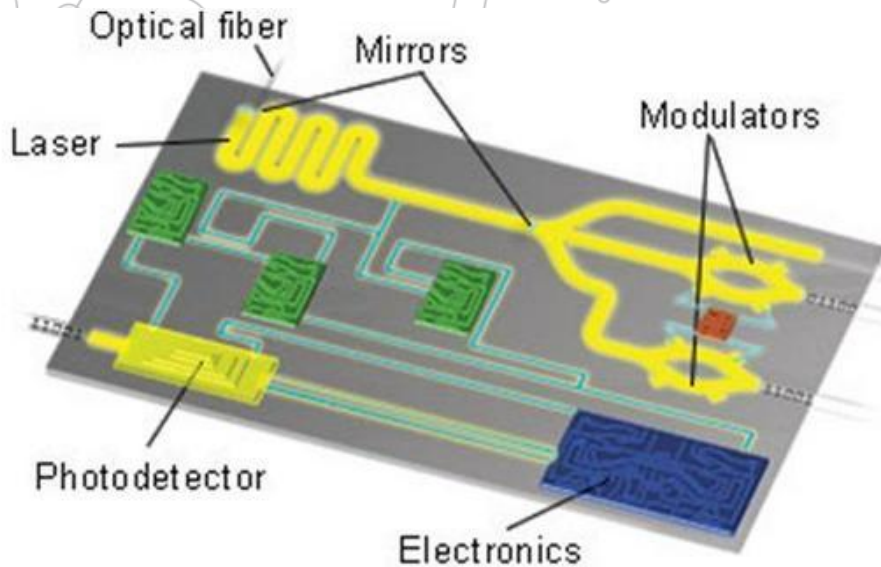
○ Baseline: no-vec for respective CPUs



icc 14.0



# Future technologies



- Performance increase in IVB-EP meets rough expectations
  - Good improvements on energy efficiency
- Core count is not the sole growth vector anymore
- Improvement in SIMD performance on Haswell
  - But our workloads don't leverage FMA extensively (partly because of compiler support)
  - Hence, for further overall performance improvement vectorizing code is essential
  - Accelerating serial code in the future will get harder and harder



# Conclusions (II)

- New territories to be explored with compiler autovectorization, OpenMP, TBB, Cilk+ and other frameworks
- Eagerly waiting for Haswell servers – will HEP workloads be ready to reap the benefits?

# Q & A