High level performance prediction following application characterization

*A. Farjallah, C. Andreolli, T. Guillet, O. Awile and P. Thierry**

Energy Engineering Team, Intel Corporation.

High level roadmap and questions

	Past (was good)	Present (excellent)	Future (even better)
Multi Cores	4S 2S 1S 1S+Gen	Standard questions: • Use of new instructions set • Impact of nb of cores • Impact of memory hierarchies, bw and latency	
Many Cores	Boot Copro	New questions: • New cores, new instructions set • Impact of nb of thread: Amdahl is back ! • Is MPI+OMP always needed. « Cluster on die » • What about GB per core • How my datacenter will look like in 2y	

Answers : applications will tell

Model application's behavior at several levels:

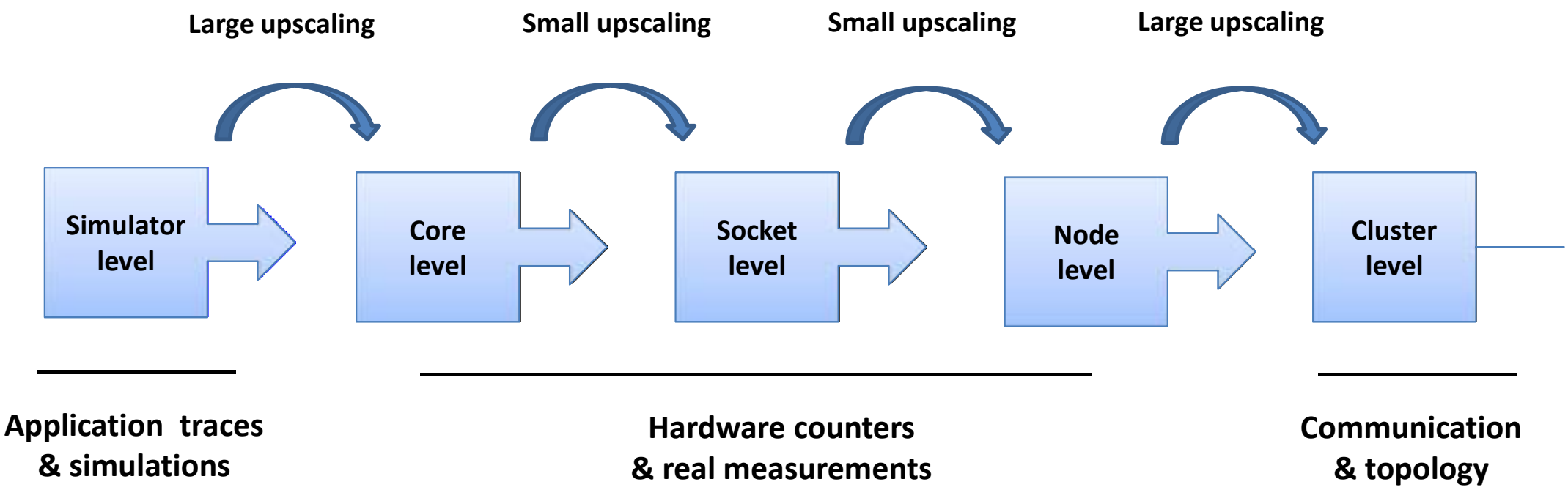
- **Determine current performance (« characterization »)**
- **Formalize an extrapolation model or use simulators**
- **Extrapolate performance on future hardware**
- **Size the future machine that will best match one or more applications**
- **Influence micro-u designs (intel internal)**

How far is this goal …

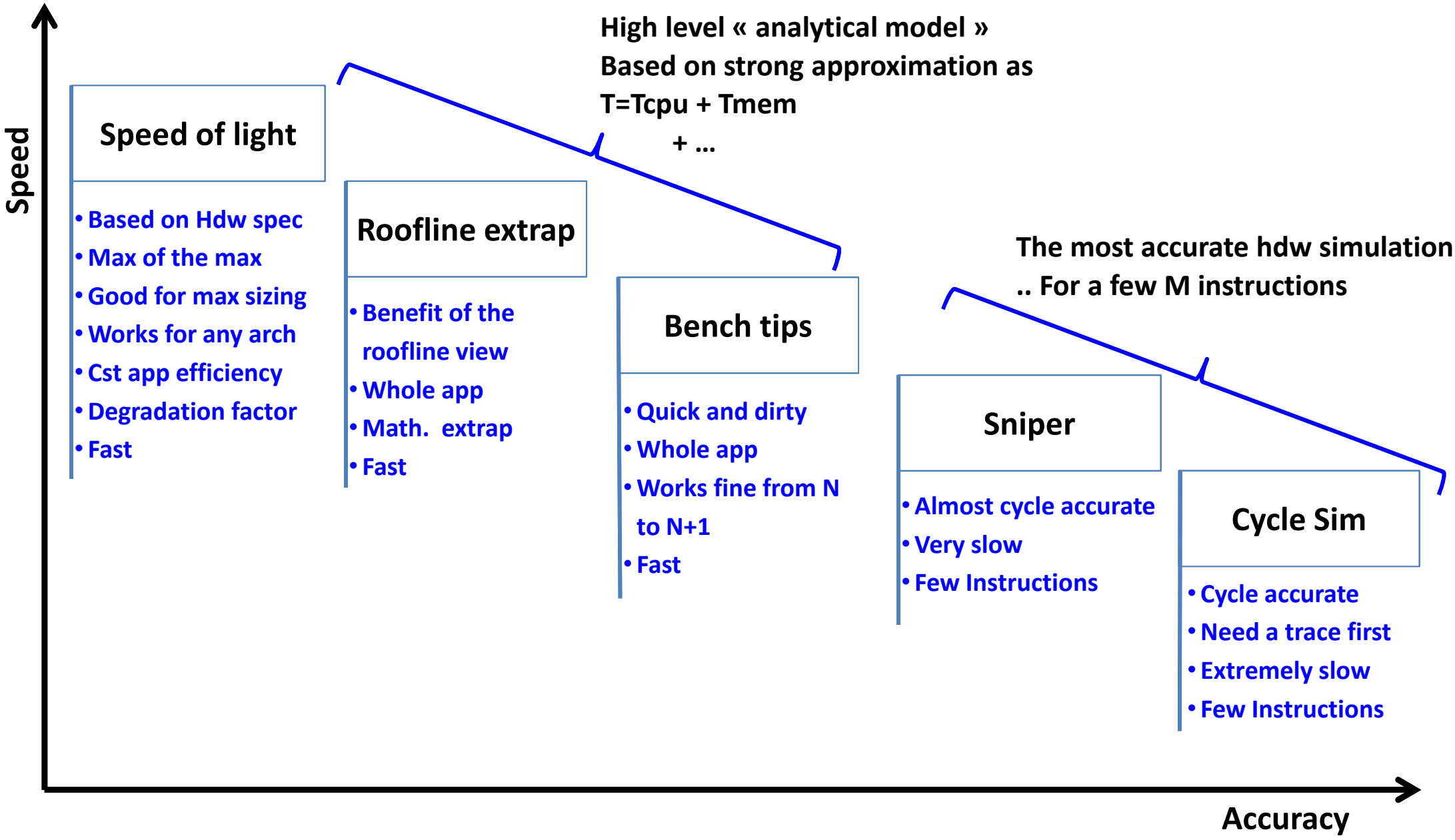
The whole model has to include every levels

from simulator to cluster level

whatever the application (implementation) is



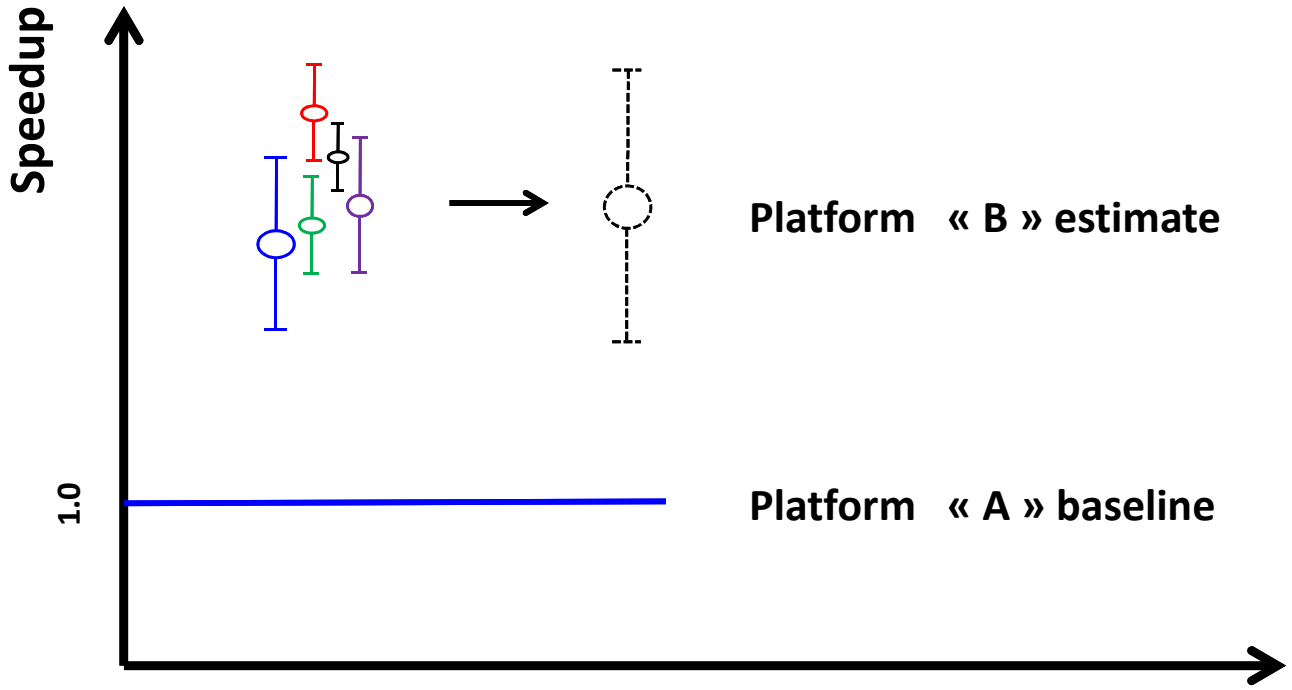
Try to be pragmatic first ..



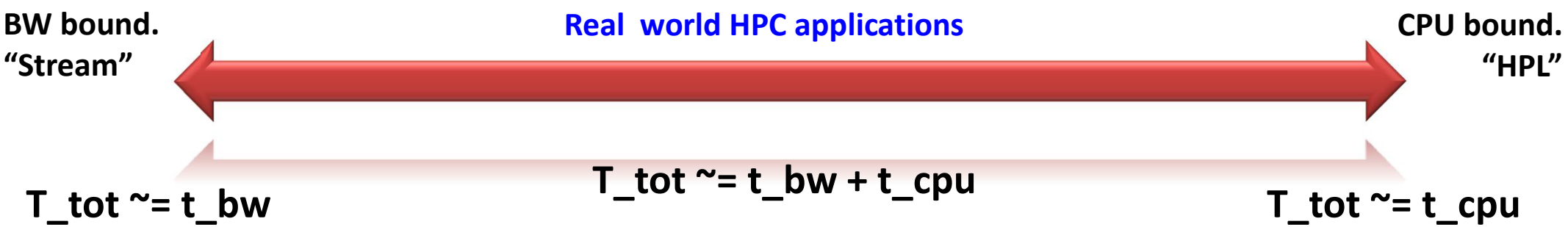
Objectives

Platform « A » → **Platform « B »**

- There is NO single answer to this problem
- Results must come from different views
- And include uncertainties



First order approximation & apps classification



Among the most important hypothesis :

- No cache nor latency impacts here
  - that may impact BW : needs higher order terms : "WIP"
- Independent Memory and CPU contributions is a wrong statement
  - More difficult to handle with analytical model
- No communication nor I/O
  - « Acceptable » if there is no huge hdw changes
  - Unacceptable if we need to model new interconnect

Performance expectation: upper bounds

BW demanding applications are bounded

by BW ratio of platforms A & B

Flops/s demanding applications are bounded

by FP ratio platforms A & B

Analyzing this ratio between 2 computers will give a first guess

defined as « speed of light»

Hypothesis: *Same efficiency on both sides (implementation, compiler, OS)*

Problem *how much to remove from this limit to account for efficiency, OS, Compiler effects ..*

« speed of light »

$$T_{tot} = t_{\_cpu} + t_{\_bw}$$

(the „cpu“ part is non correlated with the „mem“ part)

$GF_A * t_{\_cpu\_A} = GF_B * t_{\_cpu\_B}$ => so called “CPU Freq scaling” where GF denotes “the” peak FP

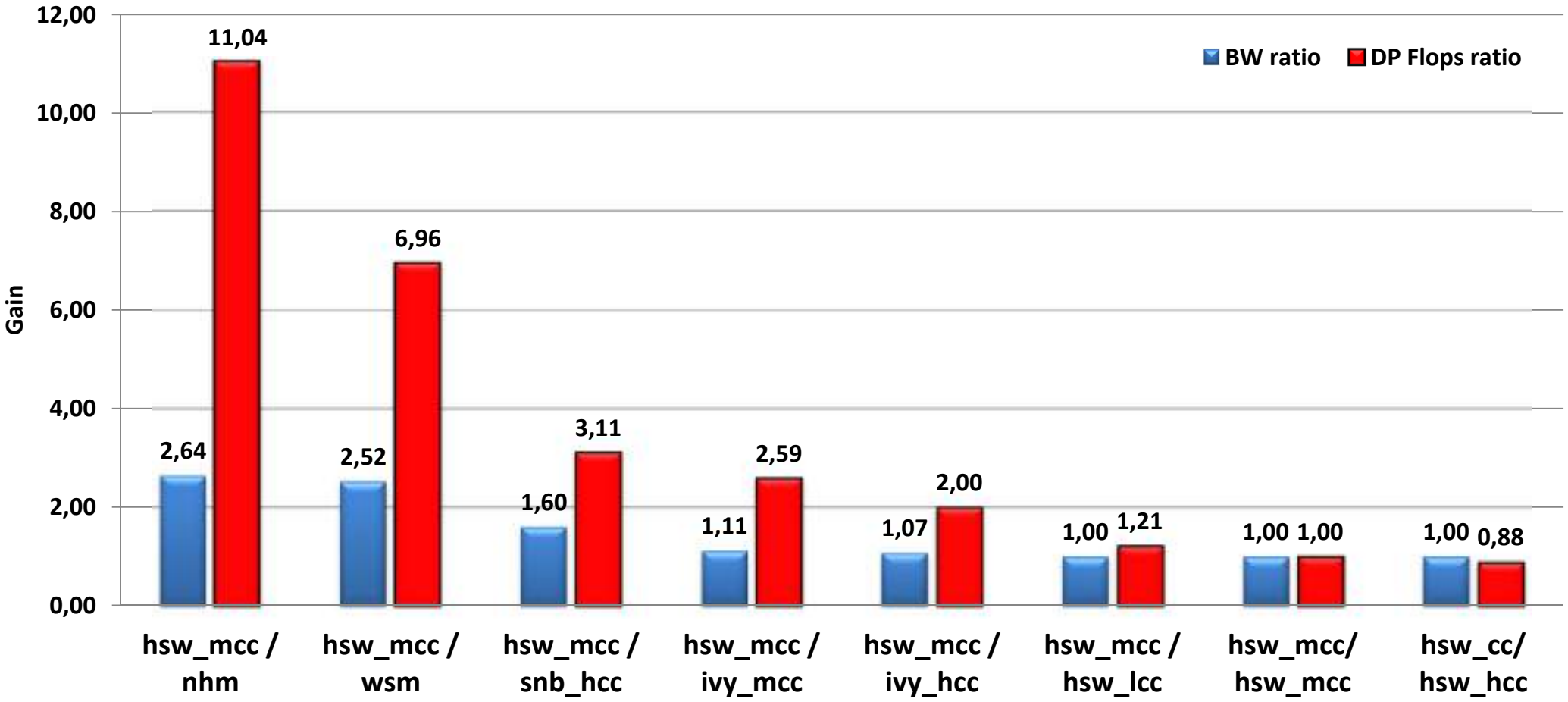
$BW_A * t_{\_bw\_A} = BW_B * t_{\_bw\_B}$ => so called “BW scaling” where BW denotes “the” peak bandwidth

The expected gain is $gain_{A \to B} = \frac{T_{tot\_A}}{T_{tot\_B}} = \frac{t_{cpu\_A} + t_{bw\_A}}{t_{cpu\_B} + t_{bw\_B}}$

For BW bounded application, $t_{\_cpu} = 0$, then $gain_{A \to B} = \frac{t_{bw\_A}}{t_{bw\_B}} = \frac{BW_B}{BW_A}$

For CPU bounded application, $t_{\_bw} = 0$, then $gain_{A \to B} = \frac{t_{cpu\_A}}{t_{cpu\_B}} = \frac{GF_B}{GF_A}$

Speed of light for HSW vs. previous micro-u



micro-u	nhm_hcc	wsm_hcc	snb_hcc	ivy_mcc	ivy_hcc	haswell_lcc	haswell_mcc	haswell_hcc
ref name		X5690	E5-2670	E5-2670v2	E5-2697v2	E5-2680 v3	E5-2697v3	E5-2699v3
instr. set	sse 4.1	sse 4.1	avx 1	avx 1	avx 1	avx2	avx2	avx2
freq	2,93	3,10	2,60	2,50	2,70	2,50	2,60	2,30
nb cores	4	6	8	10	12	12	14	18

« In between applications » : Bench tips.

Let's make 2 runs of the applications on platform A

- One with the max of cores / node
- One with half nb of cores / node (then 2x more nodes in scatter mode)

The timing can easily be split as follow:

$$T_{tot\_1} = t_{\_cpu\_A} + t_{\_bw\_A}$$

$$T_{tot\_2} = t_{\_cpu\_A} + (t_{\_bw\_A})/2$$

to obtain $t_{\_cpu\_A}$ and $t_{\_bw\_A}$ and finally the values on Platform B

$$t_{\_cpu\_B} = t_{\_cpu\_A} * (GF_A / GF_B)$$

$$t_{\_bw\_B} = t_{\_bw\_A} * (BW_A / BW_B)$$

Results

Comparison Real measurements on SNB (e5-2670) , IVY (e5-2697v2), HSW (e5-2697v3)

Simple frequency Scaling:

Extrapolation on the same micro-u (IVY / SNB) : 0. 84 %

Extrapolation on different micro-u (HSW / SNB) : - 40%

Need to extend the « CPU » contribution from simple frequency scaling using

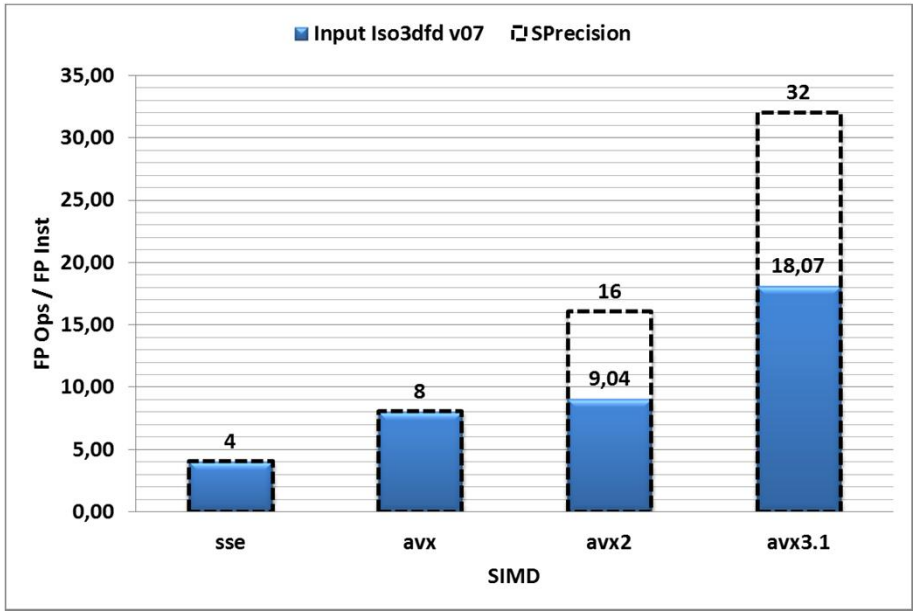
GF = (FP_ops/FP_inst)_theo * % vecto * (Inst/cyc) * (Cyc/sec) * eff * nbc

hdw SDE hdw hdw

Extrapolation on the same micro-u (IVY / SNB) : 0. 13 %

Extrapolation on different micro-u (HSW / SNB) : - 5 %

SDE to collect Instruction mix



https://software.intel.com/en-us/articles/intel-software-development-emulator

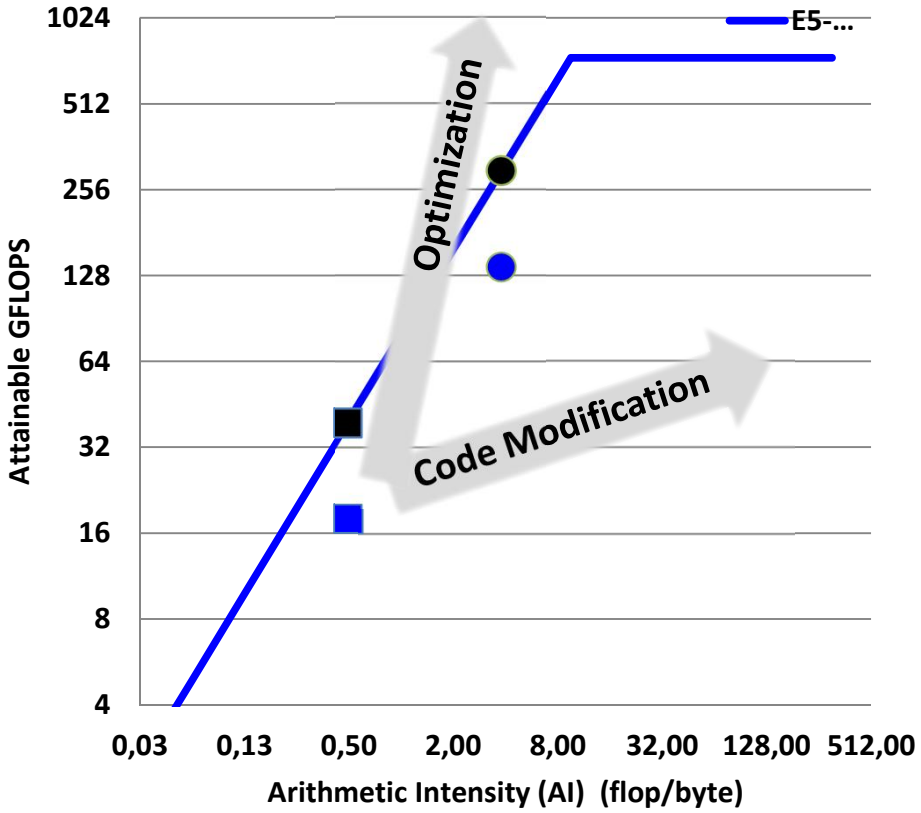
Intel icc 2015 update 1

Naïve Roofline Model

Based on Bound and Bottleneck analysis[1]

Performance is upper bounded by "a" peak flop rate and the product of "a" bandwidth and the AI

[1]D. Lazowska, J. Zahorjan, G. Graham, K. Sevcik, "Quantitative System Performance"



Gflop/s(AI) = min { xGEMM Gflop/s ; AI * StreamBW }

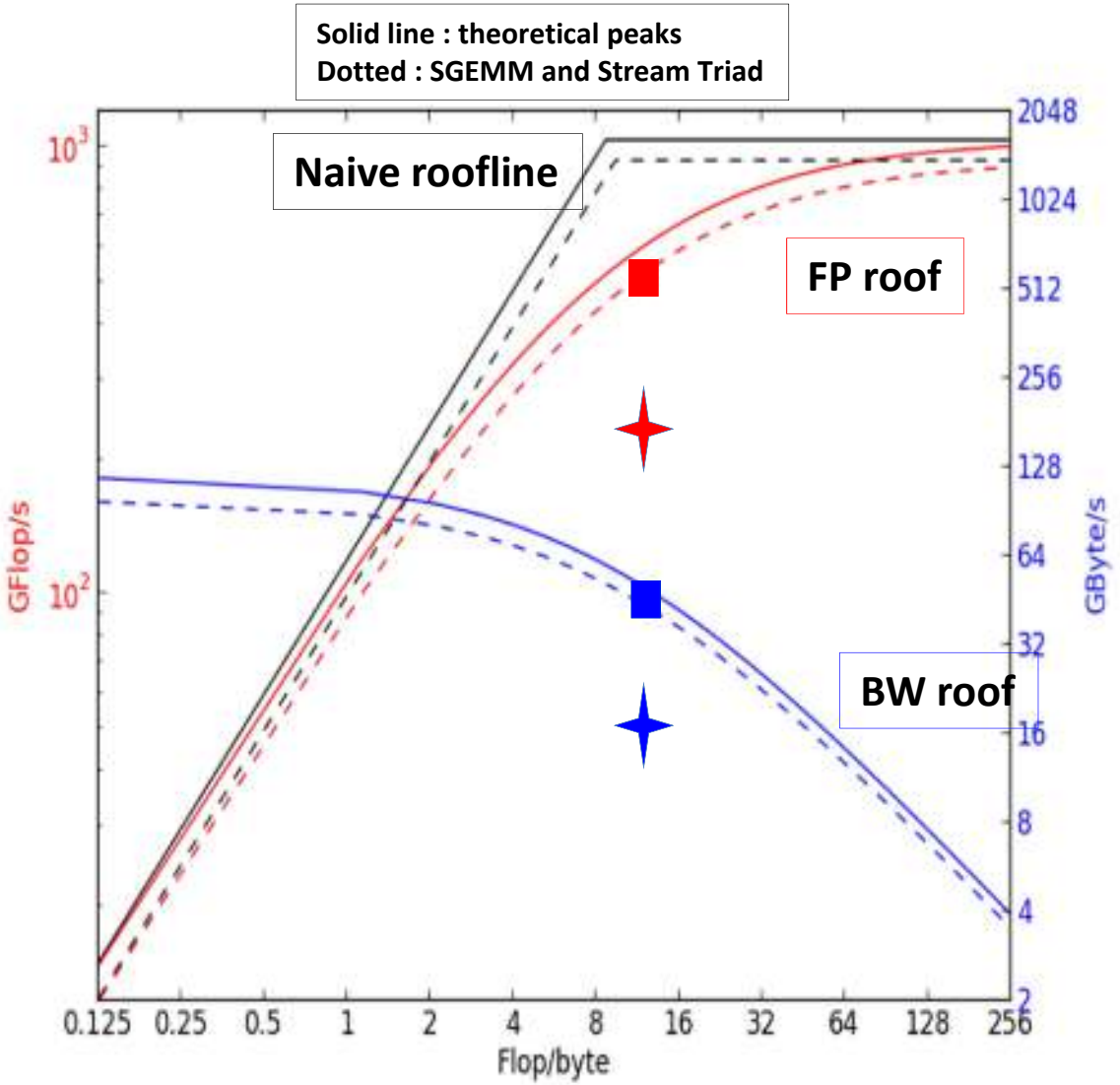
A Pair between a machine and an algorithm

<u>Know where you are before any optimization work</u>

Extended roofline $t_{tot} = t_{mem} + t_{cpu}$

$$GF / s = \frac{AI\, p_b\, p_f}{(p_f + AI * p_b)}$$

$$GB / s = \frac{p_b\, p_f}{(p_f + AI * p_b)}$$

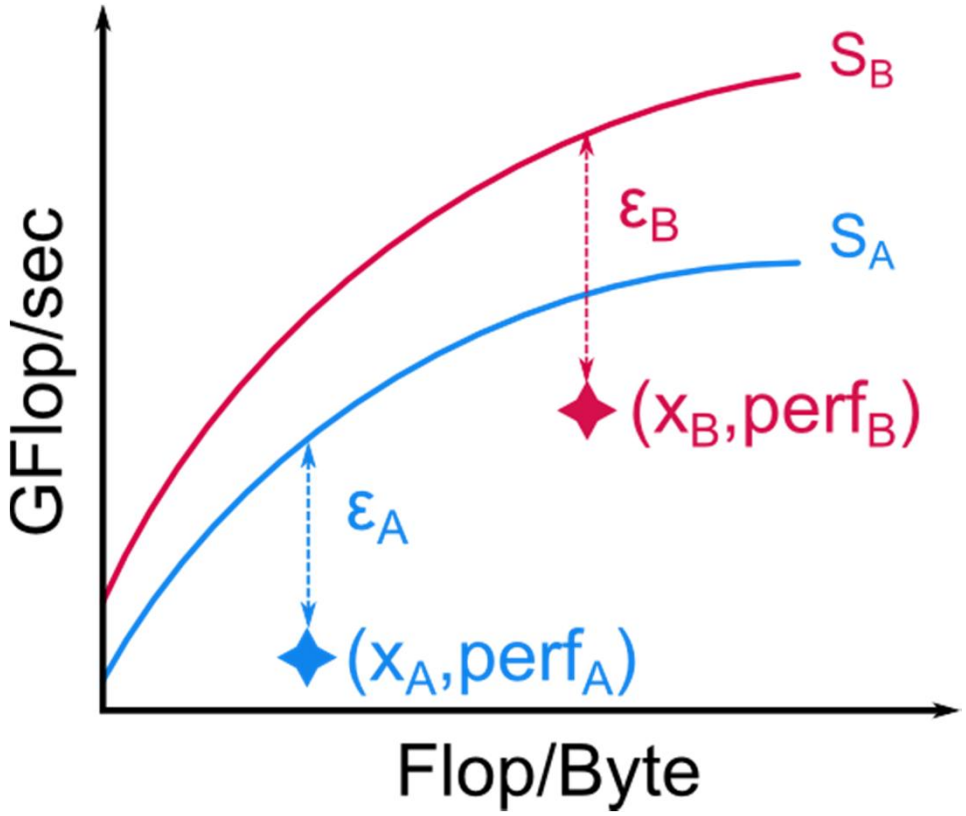


Each application should have

- an achievable peak
- a measured value

Where p_{bw} and p_f denotes bandwidth and FP peaks. Could be theoretical or measured peaks

Roofline extrapolation

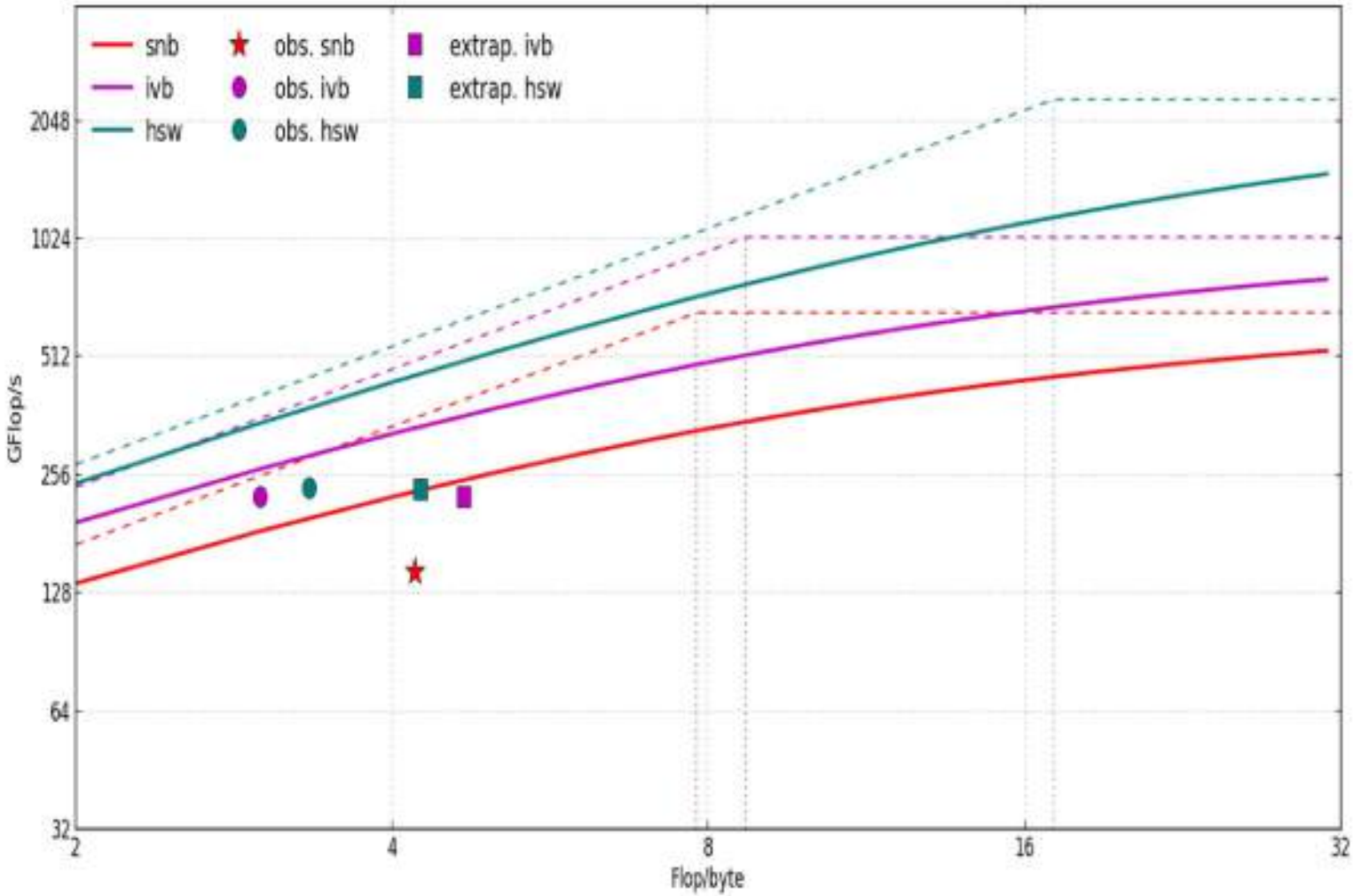


$$perf_A = \varepsilon_A \, S_A(x_A)$$
$$perf_B = \varepsilon_B \, S_B(x_B)$$

$$\Rightarrow perf_B = \varepsilon \, perf_A \frac{S_B(x_B)}{S_A(x_A)}$$

Need to formalize the « tuning » parameter with Flops and BW efficiency. WIP.

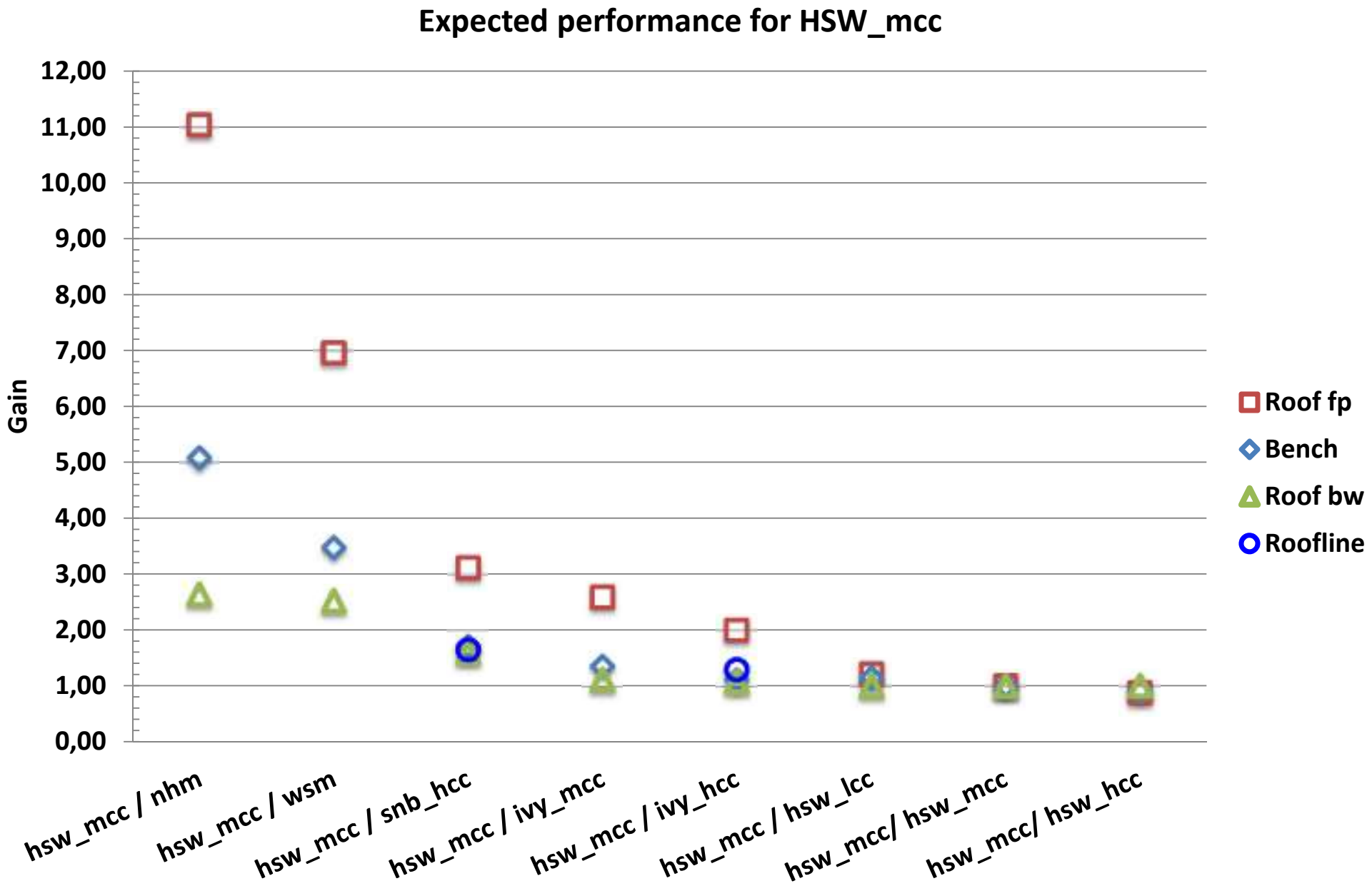
GF/s roofline extrap



Works in progress : GF and elapsed time prediction < 1% ; AI < 50%

Very similar to bench tips. Need a better definition of epsilon

High level extrapolation. Whole application



Conclusions

- Still need to master and believe in the simulator
- High level extrapolation works fine for N+2 to +3 years ahead
  - Same formalism for the 3 high levels
- Need to Include caches impact (hits, misses , latency)
  - Will increase prediction quality and range of apps
- Interconnection impacts for comms and io
  - Mandatory for cluster level
- Need to develop the uncertainties:
  - « straightforward »

Legal Disclaimers

Optimization Notice

Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel.

Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Notice revision #20110804

Legal Disclaimers

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark* and MobileMark*, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more information go to http://www.intel.com/performance.

Intel® Advanced Vector Extensions (Intel® AVX)* provides higher throughput to certain processor operations. Due to varying processor power characteristics, utilizing AVX instructions may cause a) some parts to operate at less than the rated frequency and b) some parts with Intel® Turbo Boost Technology 2.0 to not achieve any or maximum turbo frequencies. Performance varies depending on hardware, software, and system configuration and you can learn more athttp://www.intel.com/go/turbo.

Estimated Results Benchmark Disclaimer:
Results have been estimated based on internal Intel analysis and are provided for informational purposes only. Any difference in system hardware or software design or configuration may affect actual performance.

Software Source Code Disclaimer:
Any software source code reprinted in this document is furnished under a software license and may only be used or copied in accordance with the terms of that license.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.