Characterizing Applications based on Cache-Aware Roofline Model

Leonel Sousa,

Aleksandar Ilic and Diogo Marques

INESC-ID, Instituto Superior Técnico (IST),
Universidade de Lisboa, Portugal

Keynote PPAM, Bialystok, September 2019

Motivation

- **Energy-efficient architectures:** Sprinting towards Exascale Systems
  - **Heterogeneity**: compute and memory capability, specialization (CPU, GPU, big.LITTLE, FPGA)
  - Performance, power, energy and efficiency **trade-offs**

- **Application design**: Exploiting architecture diversity to reach efficiency
  - Different **computational and memory requirements**
  - **Painful** optimization and characterization (for each architecture)

- How far can we go: **Performance, Power, Energy and Efficiency Maximums?**
  - In a simple, **insightful** and fast way (allowing the first-order analysis)

Outline

RECAP: ORIGINAL ROOFLINE MODEL*

CACHE-AWARE ROOFLINE MODEL

- PERFORMANCE
- EXTENSIONS: POWER AND ENERGY-EFFICIENCY

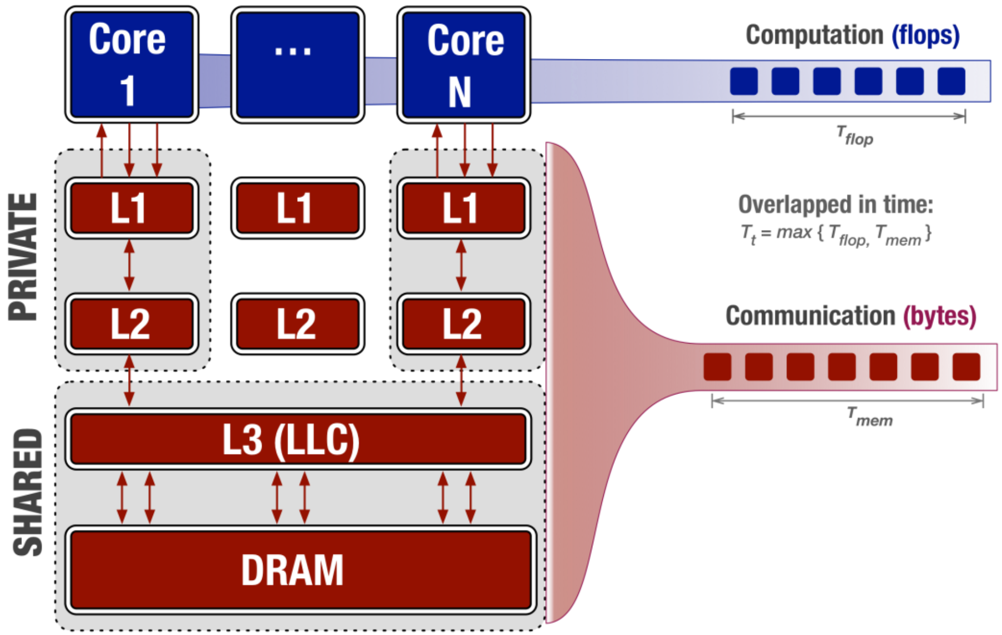
APPLICATION CHARACTERIZATION WITH CACHE-AWARE ROOFLINE MODEL

- APPLICATION-DRIVEN CARM
- ISO3DFD CASE STUDY
- PROXY-APPLICATIONS FROM EXASCALE COMPUTING PROJECT
- BAYESIAN K2 SCORE

ON-GOING (FUNDED) PROJECTS

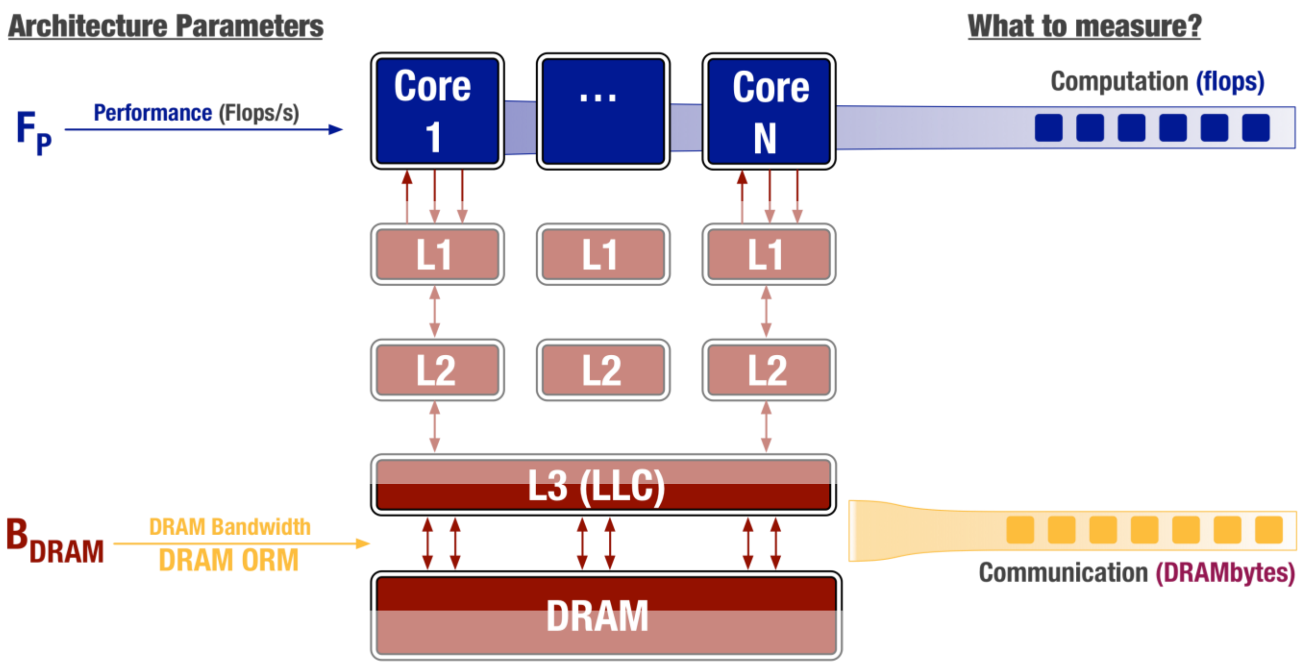
** Williams S., et. al., "Roofline: An insightful visual performance model for multicore architectures", CACM (2009)*

Multi-cores and the Roofline Model

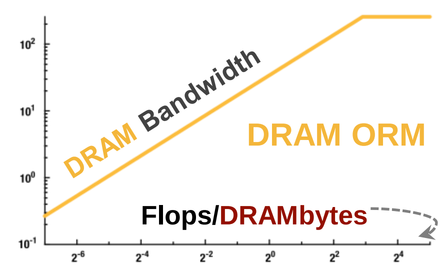


- General-purpose processors with complex memory hierarchy
  - **Multiple cores** with powerful out-of-order engines
  - **Several levels of memory hierarchy:** private/shared caches + deeper (and diverse) memory levels
- **Observation:** Computations and communication (data transfers) simultaneously performed
  - The overall execution time can be limited either by the time to compute or by the time taken to transfer data
  - **Different Roofline Models observe memory traffic differently!**

DRAM Roofline Model



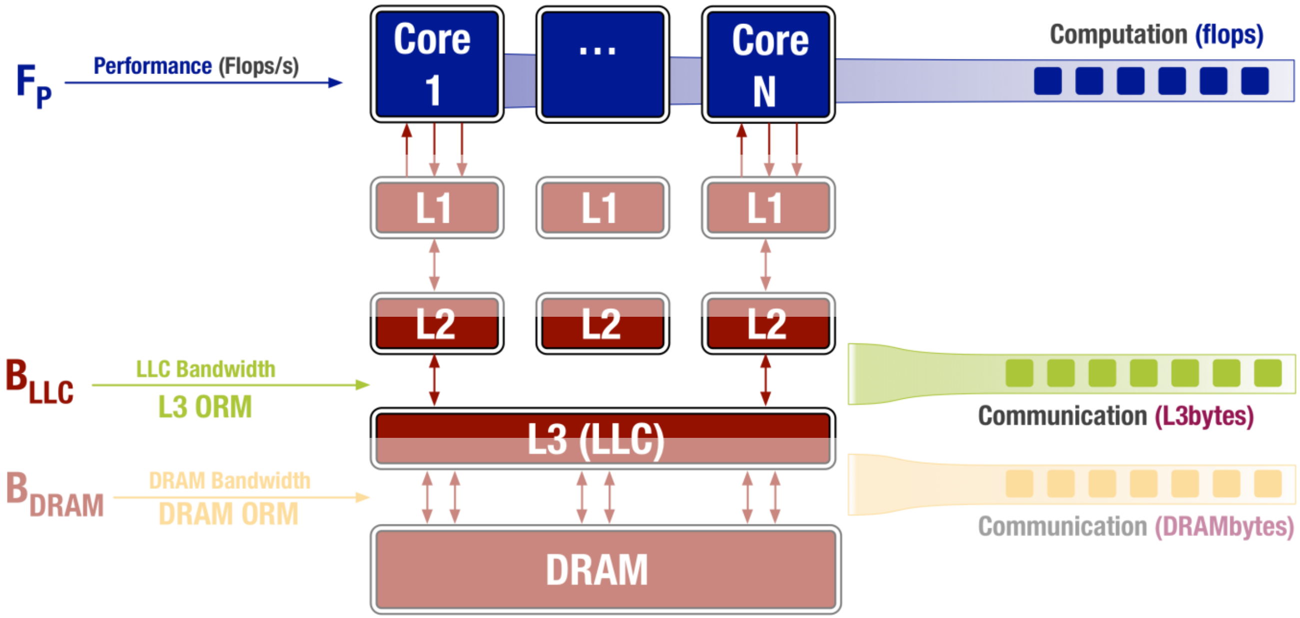
- **DRAM ORM:** DRAM-based Original Roofline Model
  - **Memory traffic: bytes between DRAM and LLC**
  - Memory Bandwidth: DRAM to LLC, i.e., **DRAM bandwidth**
  - Compute performance: Flops delivered by the core(s)
  - **Intensity (x-axis): Flops/DRAMBytes**

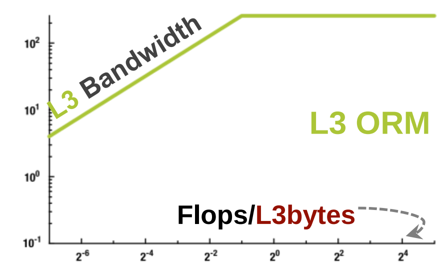


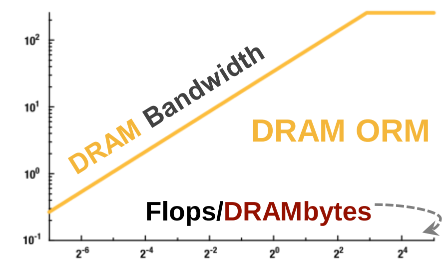
Original Roofline Model

Architecture Parameters

What to measure?

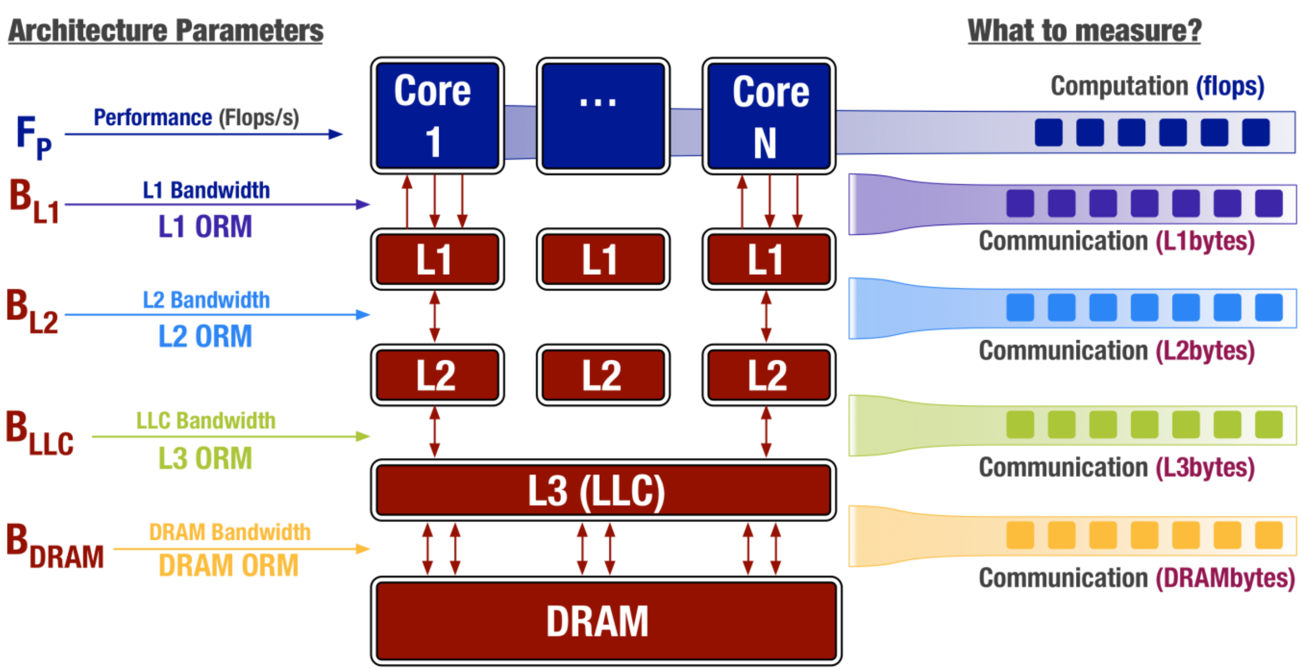


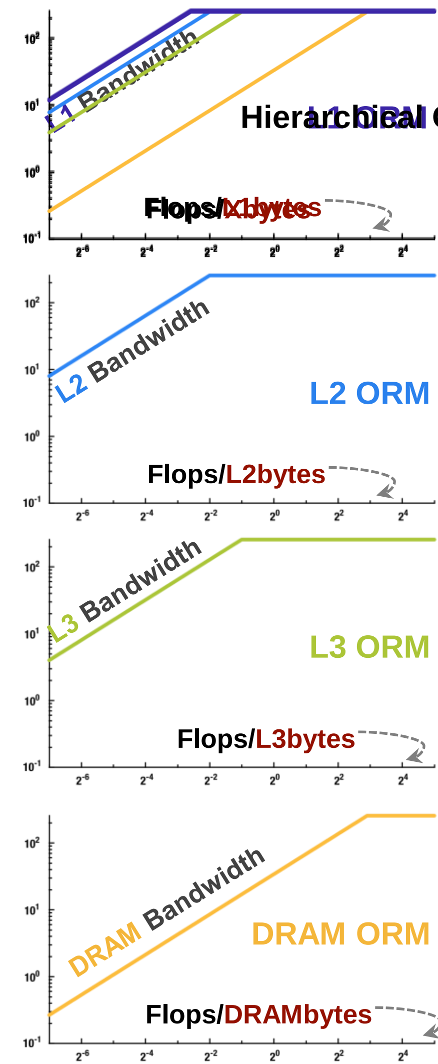




- **L3 ORM:** L3-based Original Roofline Model
  - **Memory traffic: bytes between LLC and L2**
  - Memory Bandwidth: LLC to L2, i.e., **L3 bandwidth**
  - Compute performance: Flops delivered by the core(s)
  - **Intensity (x-axis): Flops/L3Bytes**

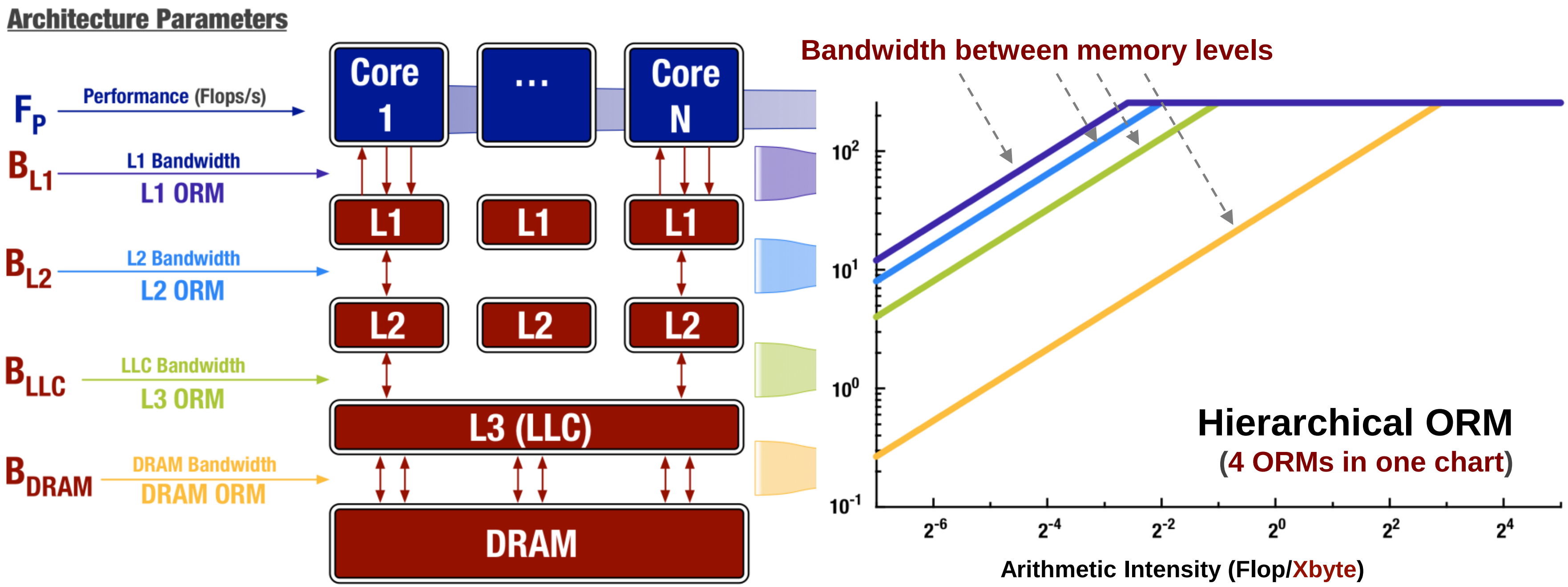
Original Roofline Model





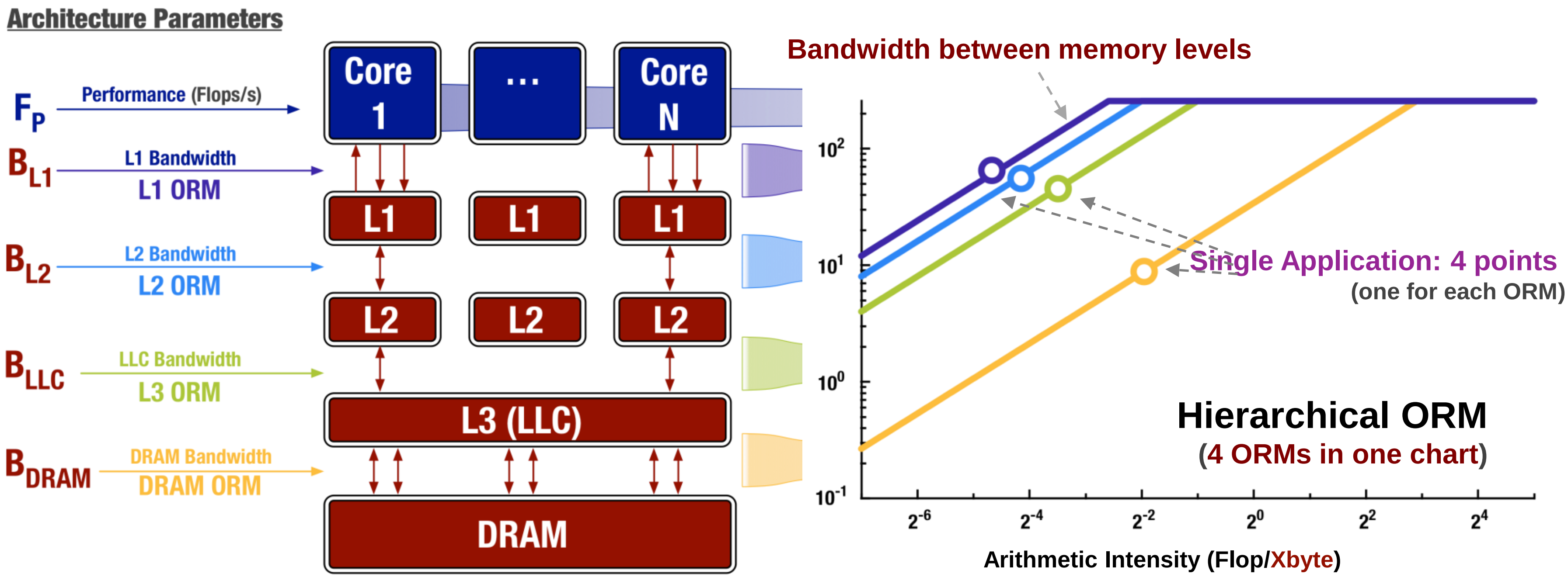
- **Lx ORM:** Lx-based Original Roofline Model
  - **Memory traffic: bytes between two subsequent memory levels**
  - Memory Bandwidth: **Lx bandwidth**
  - Compute performance: Flops delivered by the core(s)
  - **Intensity (x-axis): Flops/LxBytes**

Hierarchical ORM



- **Hierarchical ORM:** Several Lx-based Original Roofline Models in a single plot
  - Memory Bandwidth: **Lx bandwidth (for each ORM)**
  - Compute performance: Flops delivered by the core(s)
  - **Intensity (x-axis): Flops/LxBytes**

Hierarchical ORM



- **Hierarchical ORM:** Several Lx-based Original Roofline Models in a single plot
  - Memory Bandwidth: **Lx bandwidth (for each ORM)**
  - Compute performance: Flops delivered by the core(s)
  - **Intensity (x-axis): Flops/LxBytes**

  - **Application characterization: As many points as memory levels (one for each ORM)**

Outline

RECAP: ORIGINAL ROOFLINE MODEL*

CACHE-AWARE ROOFLINE MODEL

- PERFORMANCE*
- EXTENSIONS: POWER AND ENERGY-EFFICIENCY

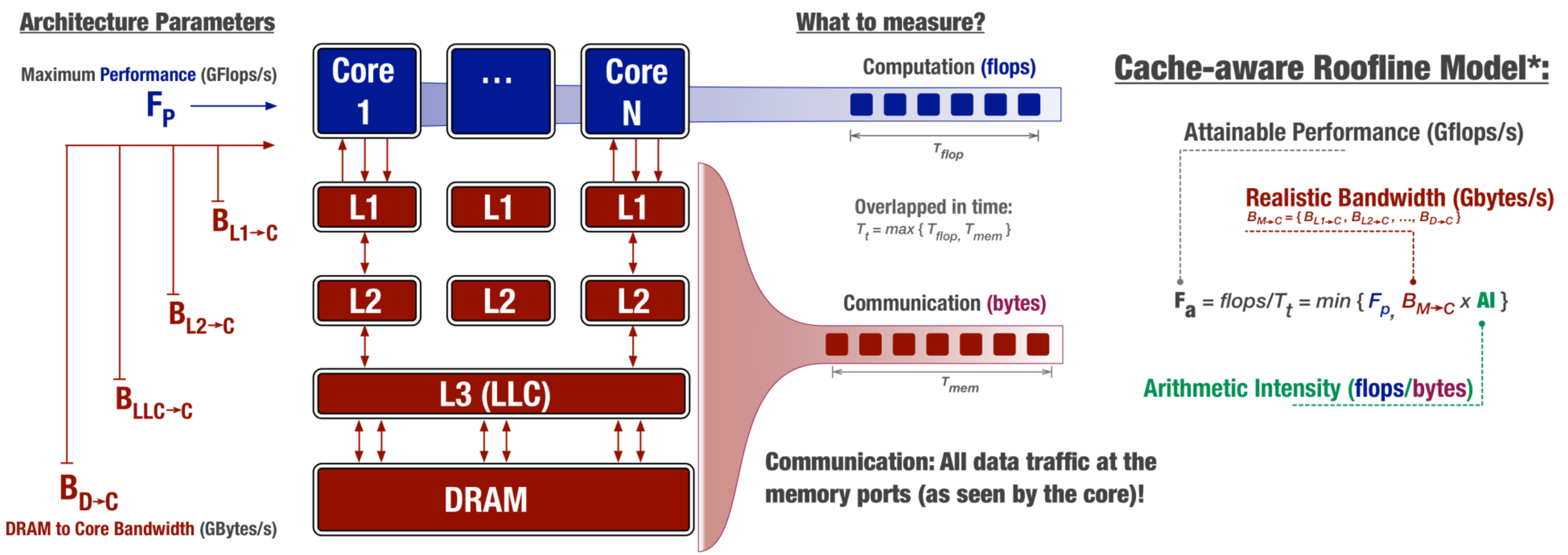
APPLICATION CHARACTERIZATION WITH CACHE-AWARE ROOFLINE MODEL

- APPLICATION-DRIVEN CARM
- ISO3DFD CASE STUDY
- PROXY-APPLICATIONS FROM EXASCALE COMPUTING PROJECT
- BAYESIAN K2 SCORE

ON-GOING (FUNDED) PROJECTS

A. Ilic, F. Pratas and L. Sousa "Cache-ware Roofline Model: Upgrading the Loft", IEEE Computer Architecture Letters (2014)

Cache-aware Roofline Model (CARM)

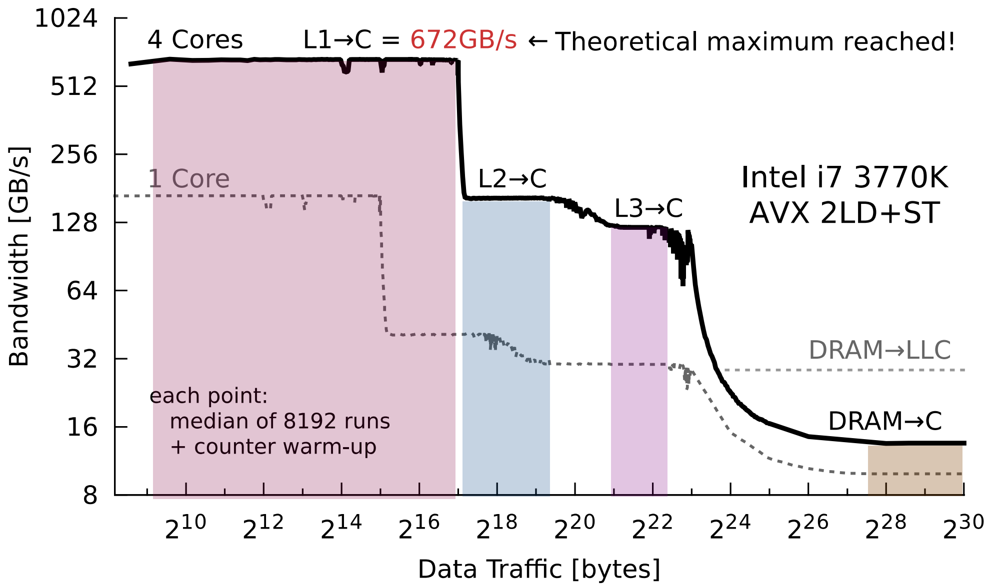


- CARM: visual representation of the **limits of parallel processing** on contemporary multi-cores
  - Considers memory traffic and computation from the **consistent architecture point of view** (cores)
  - Relates the peak compute performance and realistically achievable bandwidth with **Arithmetic Intensity (AI)**
  - Unifies the complete memory hierarchy in a **single plot model**
- **CONSTRUCTION**: How to obtain these bandwidth values? (only $B_{L1\text{->}C}$ directly derivable from data sheets)

* *Ilic A., Pratas F., Sousa L., “Cache-aware Roofline Model: Upgrading the loft”, IEEE CAL (2014)*

Developed Tools: Methodology and micro-benchmarks

- **CARM-oriented tools:** construction, validation and application characterization
  - **SchedMon**[1]: Software tool for near-OS counter-based monitoring (multiplexing and shed events)
  - **LARM**[2]: CARM bandwidth and FP performance micro-benchmarks (NUMA/KNL CARM)
  - **CARM** validation tools and micro-benchmarks[3]

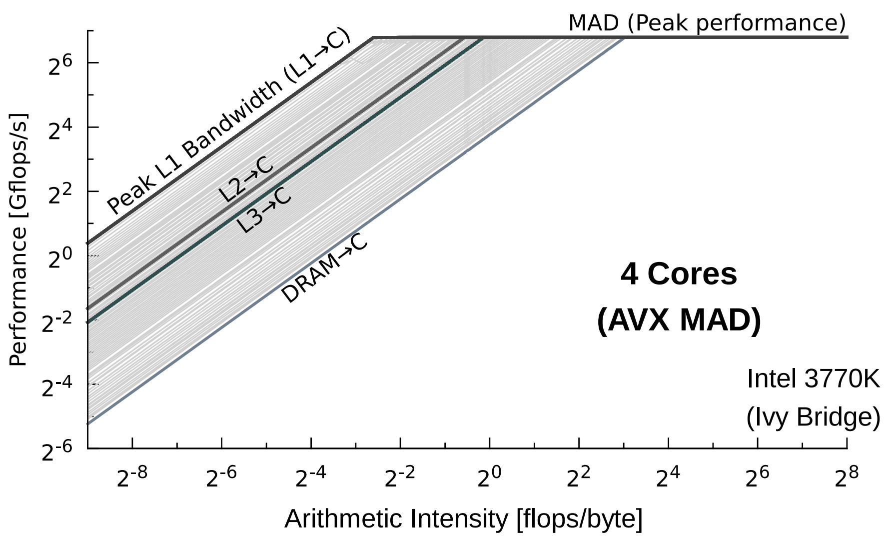



```
// Configured Counters
CPU_CLK_UNHALTED.CORE/REF
MEM_UOP_RETIRED.ALL_LOADS
MEM_UOP_RETIRED.ALL_STORES
…

// AVX Assembly: 2LD+1ST
vmovapd   0(%rax), %ymm0
vmovapd  32(%rax), %ymm1
vmovapd     %ymm2, 64(%rax)
vmovapd  96(%rax), %ymm3
vmovapd 128(%rax), %ymm4
vmovapd     %ymm5, 160(%rax)
…
```

[1] *Taniça L., et. al. "SchedMon: A Performance and Energy Monitoring Tool for Modern Multi-cores", MuCoCoS/Euro-Par (2014)*
[2] *Denoyelle N., et. al., "Modeling Large Compute Nodes with Heterogeneous Memories in CARM", PMBS, SC (2017) and IEEE TPDS (2018)*
[3] *Marques D., "Analyzing Performance of Multi-cores and Applications with Cache-aware Roofline Model", HPBench, HPCS (2017)*

Cache-aware Roofline Model

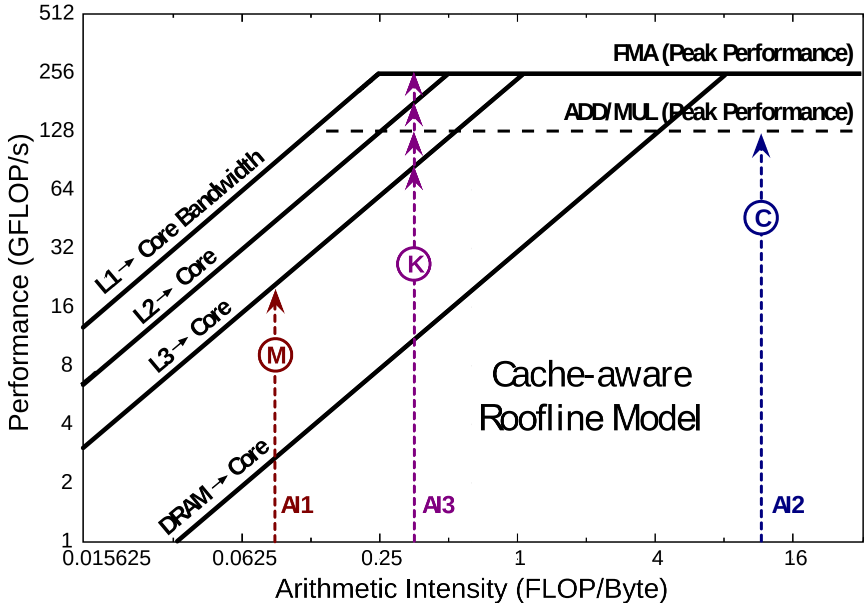


- **Total Cache-aware Roofline Model**
  - Includes **all transitional states** (traversing the memory hierarchy and filling the pipeline)
  - Single-plot modeling for **different** types of compute and memory **operations**

- Insightful **single plot model**
  - Shows performance limits of multicores
  - Redefined AI: flops and bytes as seen by core
  - Constructed once per architecture
- Considers **complete memory hierarchy**
  - Influence of caches and DRAM to performance
- Applicable to **other types of operations**
  - not only floating-point
- **Useful for:**
  - **Application** characterization and optimization
  - **Architecture** development and understanding

** Ilic A., Pratas F., Sousa L., "Cache-aware Roofline Model: Upgrading the loft", IEEE CAL (2014)*

Cache-aware Roofline Model: Interpretation



- **Performance analysis with CARM**
  - **Application** (kernel) is a **single "dot"**
  - In respect to their AI and FP Performance

- Draw an **imaginary vertical line** at app AI
  - Arithmetic intensity: Property of the application
  - Should not change (unless the algorithm changes)

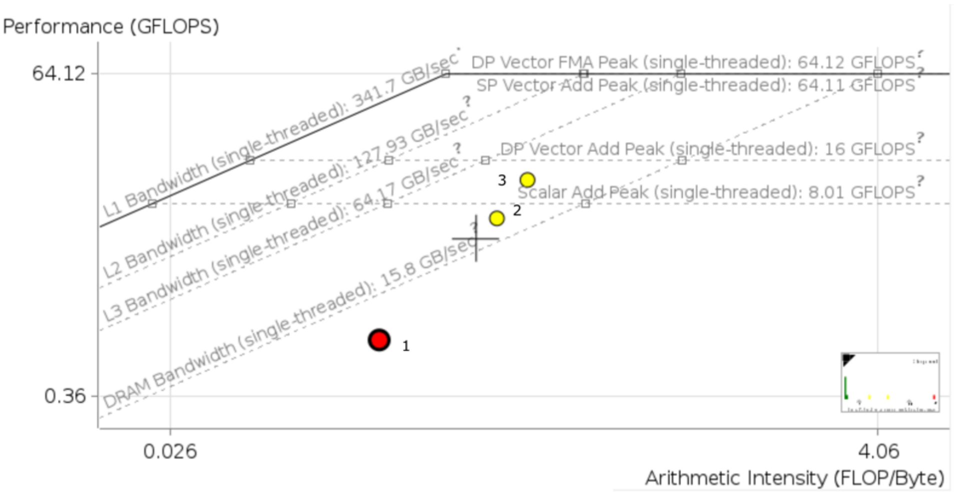
- Intersected roofs: **potential bottlenecks**
  - Priority to the roofs above
  - Roofs below are also important!

- **Optimization hints**
  - **Memory**: improve access pattern, use of caches
  - **Compute**: vectorization, FMAs, parallelization
  - **Shady**: all kinds of everything (mem+comp)

- **Optimization: "Break the above roofs"**
  - Optimizations should improve the performance
  - Points move up in the CARM plot

** Marques D., et. al., "Performance Analysis with Cache-Aware Roofline Model in Intel Advisor", HPCS (2017)*

Cache-aware Roofline Model in Intel Advisor



Advisor Cache-aware Roofline

- **Automatic construction** (from NHL to KBL-X)
- Break-down by application phases, loops and functions (hierarchical feature)
- In-depth **application profiling and optimization** hints

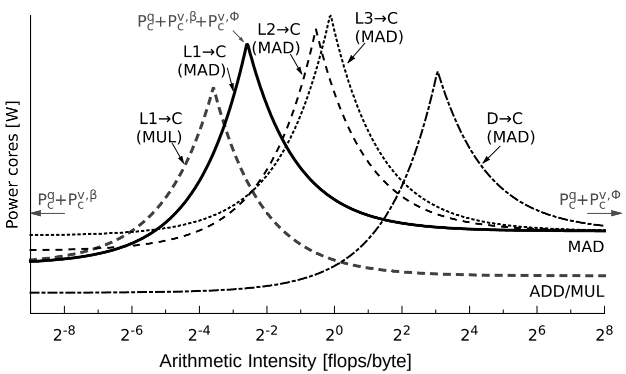
- **Performance analysis with CARM**
  - Applications (kernels) as single "dots"
  - In respect to their AI and FP Performance
- Draw an **imaginary vertical line** at app AI
  - Arithmetic intensity: Property of the application
  - Should not change (unless the algorithm changes)
- Intersected roofs: **potential bottlenecks**
  - Priority to the roofs above
  - Roofs below are also important!
- **Optimization: "Break the above roofs"**
  - Optimizations should improve the performance
  - Points move up in the CARM plot

"incredibly useful diagnosis tool (that can guide the developers in the application optimization process), ensuring that they can squeeze the maximum performance out of their code with minimal time and effort."

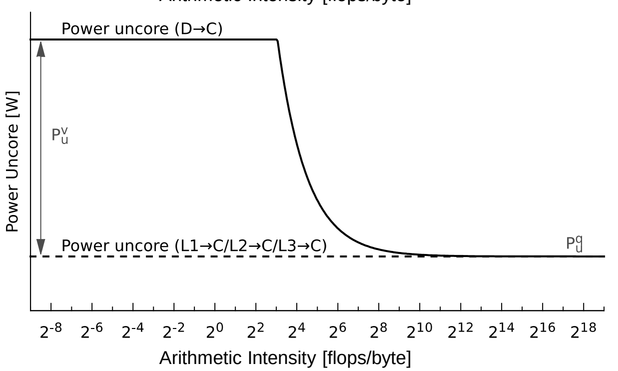
Cache-aware Roofline Model: Power Consumption

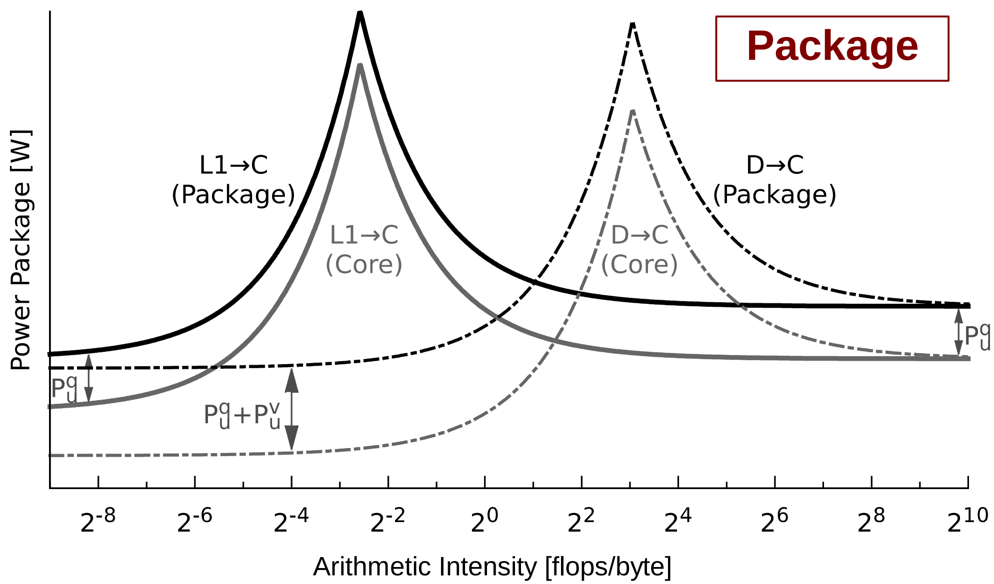
- **Performance**: Computations (*flops*) and communication (*bytes*) overlap in time
- **Power consumption**: *Superposed* contributions of *flops* and *bytes*

Cores



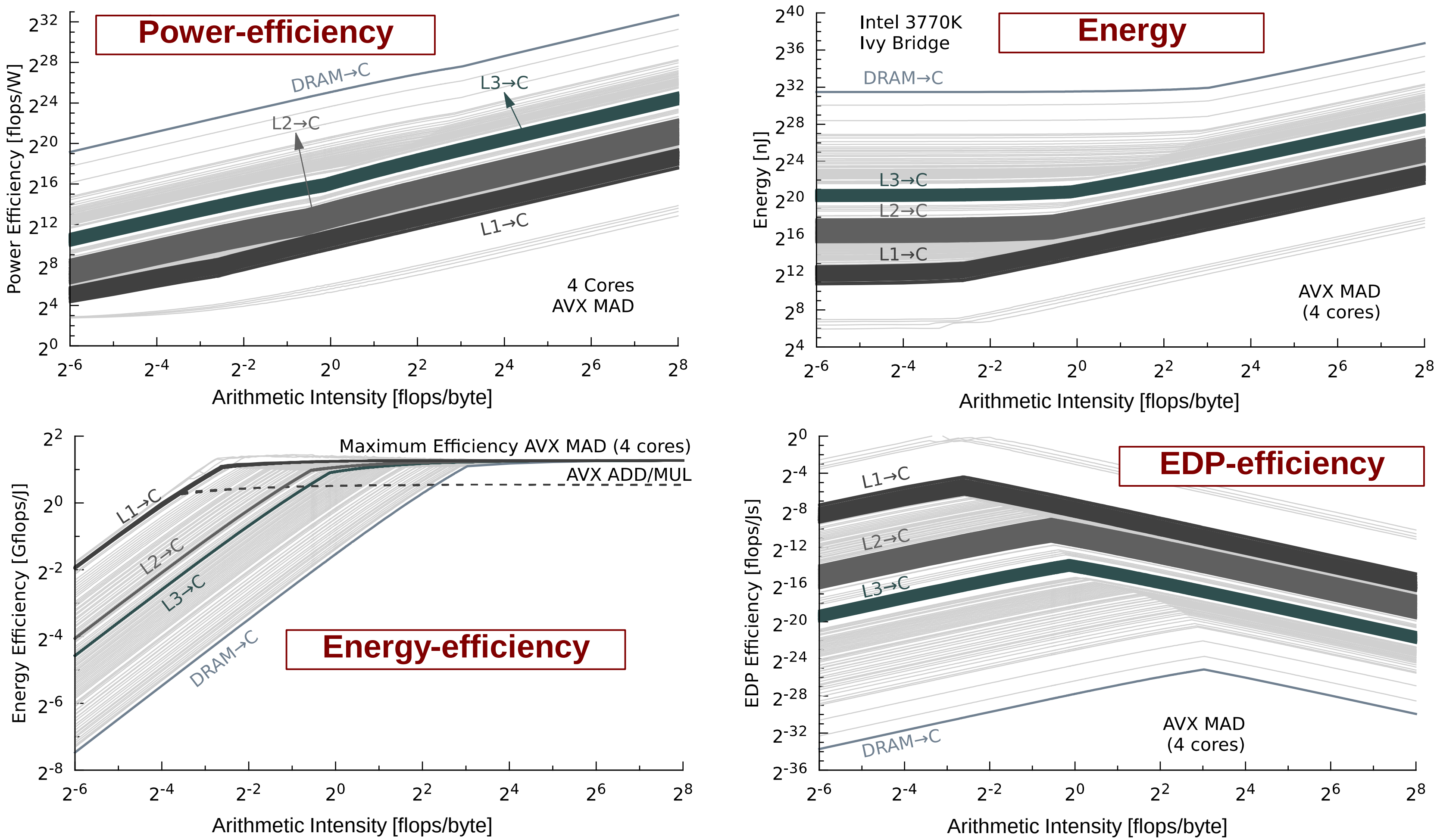
Uncore





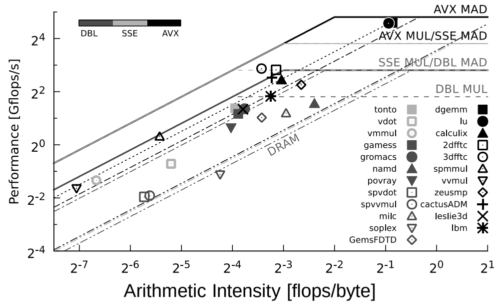
* Ilic, A., Pratas, F. and Sousa, L., “Beyond the Roofline: Cache-aware Power & Energy-Efficiency Modeling…”, IEEE Transactions on Computers (2016)

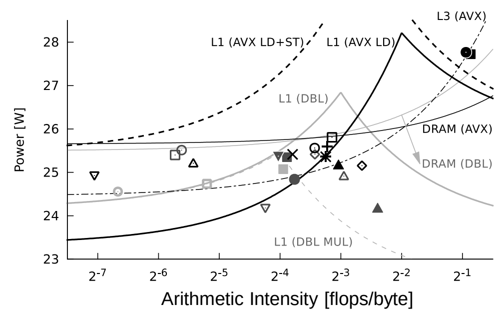
Efficiency CARMs

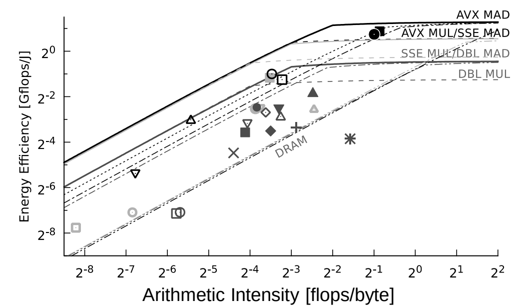


Cache-aware Roofline Model: Use Cases

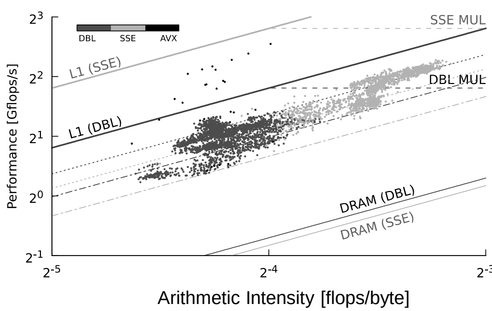
Application Characterization

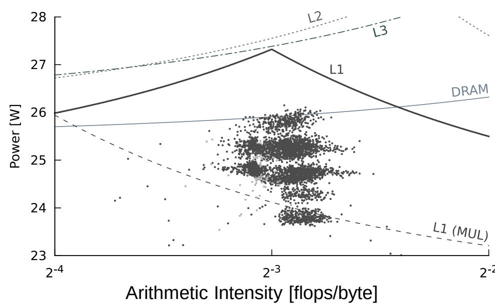


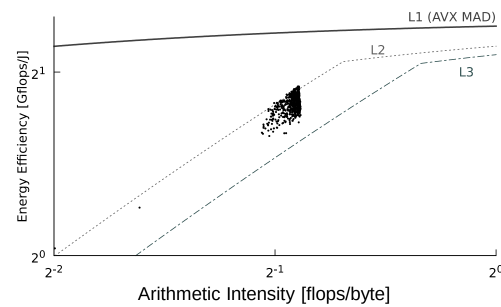




Online Monitoring



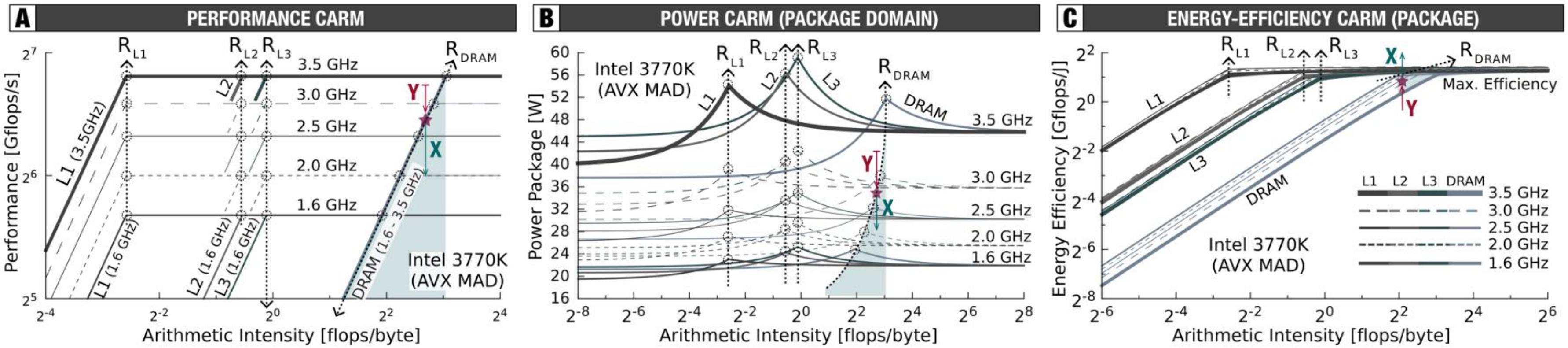




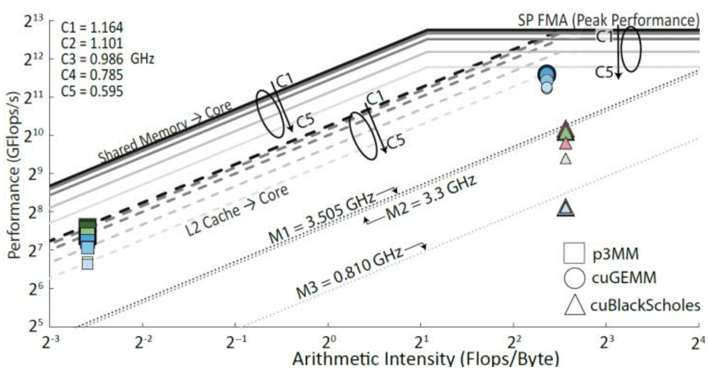
* *Ilic, A., Pratas, F. and Sousa, L., "Beyond the Roofline: Cache-aware Power & Energy-Efficiency Modeling…", IEEE Transactions on Computers (2017)*
* *Antão, D., et.al.,"Monitoring Performance and Power for Application Characterization with CARM", PPAM'13*

Cache-Aware Roofline Model: Extensions

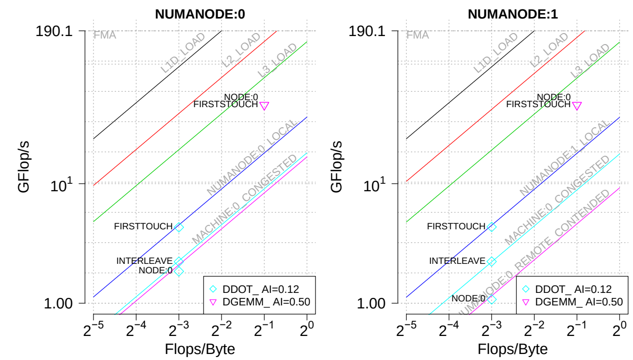
CARM-based DVFS analysis



GPU CARM



NUMA CARM



* *Ilic A., et.al., "Beyond the Roofline: Cache-aware Power & Energy-Efficiency…", IEEE Transactions on Computers (2017)*
* *Lopes A., et.al, "Exploring GPU performance, power and energy-efficiency bounds with CARM"", ISPASS (2017)*
* *Denoyelle N., et.al., "*Modeling Non-Uniform Memory Access on Large Compute Nodes with the Cache-Aware Roofline Model*", IEEE Transactions on Parallel and Distributed Systema (2018)*

Summary

SEVERAL CACHE-AWARE ROOFLINE MODELS (EXPERIMENTALLY VERIFIED)

– **(Total) Performance CARM**
– **(Total) Power CARM**: for several domains, i.e., power of cores, uncore power and complete package power
– **Energy-Efficiency CARM**: Performance + Power Domains
– **Energy, Power-efficiency and EDP-based CARMs**
– **DVFS, GPU and NUMA CARMs**

ON-GOING WORK

- CARM FOR ARM, FPGAS, COMPLETE SYSTEM …

Outline

RECAP: ORIGINAL ROOFLINE MODEL*

CACHE-AWARE ROOFLINE MODEL

- PERFORMANCE*
- EXTENSIONS: POWER AND ENERGY-EFFICIENCY

APPLICATION CHARACTERIZATION WITH CACHE-AWARE ROOFLINE MODEL

- APPLICATION-DRIVEN CARM
- ISO3DFD CASE STUDY
- PROXY-APPLICATIONS FROM EXASCALE COMPUTING PROJECT
- BAYESIAN K2 SCORE

ON-GOING (FUNDED) PROJECTS

**A. Ilic, F. Pratas and L. Sousa "Cache-ware Roofline Model: Upgrading the Loft", IEEE Computer Architecture Letters (2014)*
Recently submitted publication

Application-driven CARM

Applications
Can they exploit the modeled absolute maximums?

- Large set of phases with **diverse characteristics**
- Different instruction mix, vectorization, SP/DP and LD/ST balance, FP share, memory access pattern…

Hardware
Do the Rooflines reflect the application demands?

- Maximums vary with **utilization/execution scenario**
- **Components/subsystems** differently exercised: ports, compute units, front-end, back-end, sockets …
- **Memory** subsystem: deep and diverse hierarchy, caches (private/shared), DRAM, NUMA, HBM…

Insightful Micro-architecture Modeling
Existing approaches model the absolute maximums

- Disjoint Roofline Methodologies
  (*Cache-aware, Classic ORM, Hierarchical, Integrated, …*)
- May provide misleading optimization guidelines
- Inconclusive bottleneck detection

Application-driven CARM

Applications
Can they exploit the modeled absolute maximums?

- Large set of phases with **diverse characteristics**
- Different instruction mix, vectorization, SP/DP and LD/ST balance, FP share, memory access pattern…

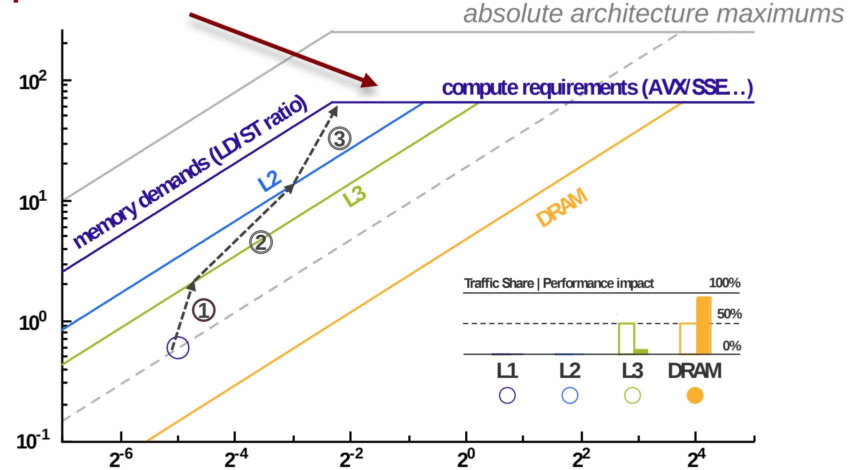
Hardware
Do the Rooflines reflect the application demands?

- Maximums vary with **utilization/execution scenario**
- **Components/subsystems** differently exercised: ports, compute units, front-end, back-end, sockets …
- **Memory** subsystem: deep and diverse hierarchy, caches (private/shared), DRAM, NUMA, HBM…

Insightful Micro-architecture Modeling
Existing approaches model the absolute maximums

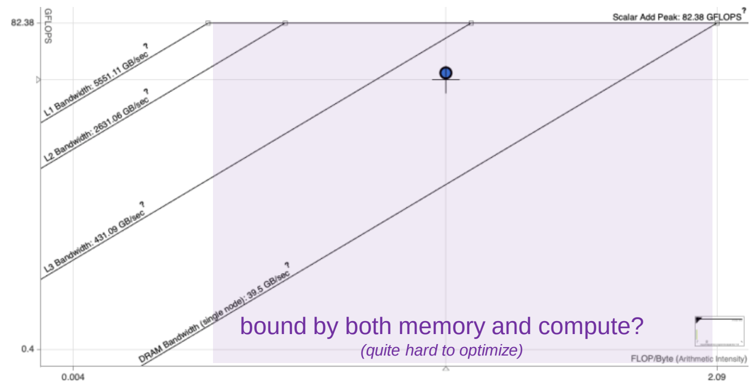
- Disjoint Roofline Methodologies
  (Cache-aware, Classic ORM, Hierarchical, Integrated,…)
- May provide misleading optimization guidelines
- Inconclusive bottleneck detection

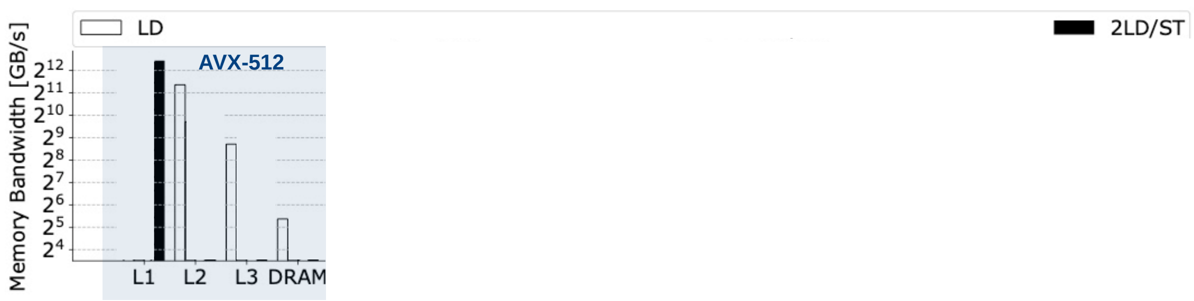
Application-driven Roofline



Application-driven CARM

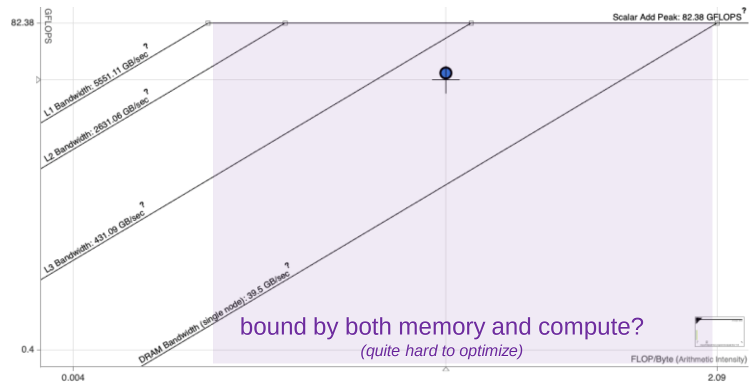
State-of-the-art CARM

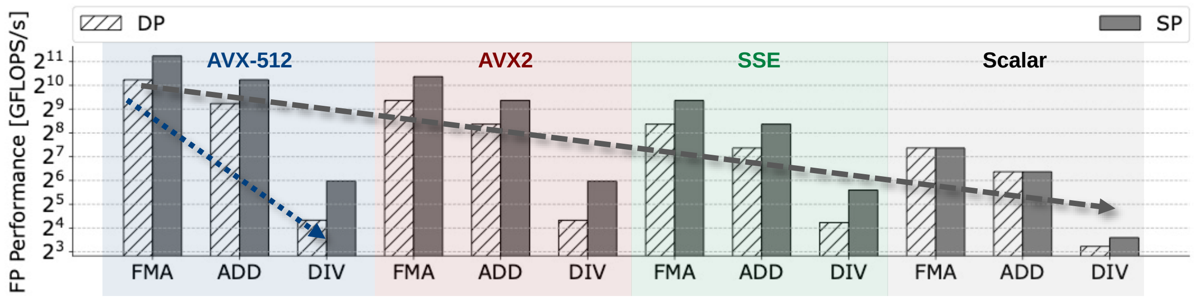


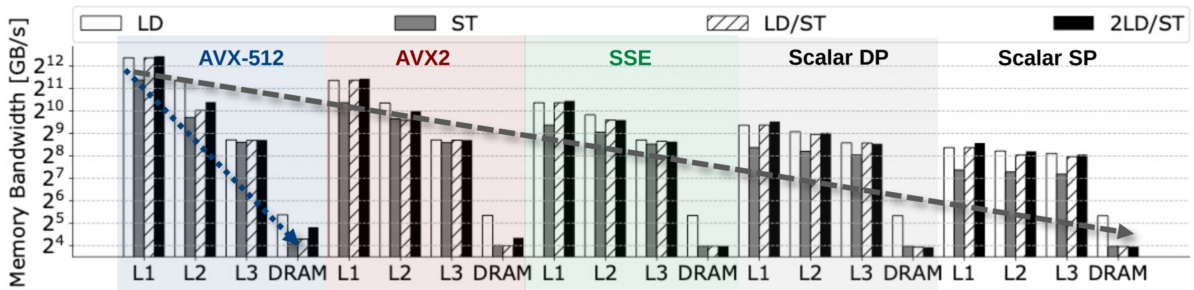


Application-driven CARM (SKL-X)

State-of-the-art CARM





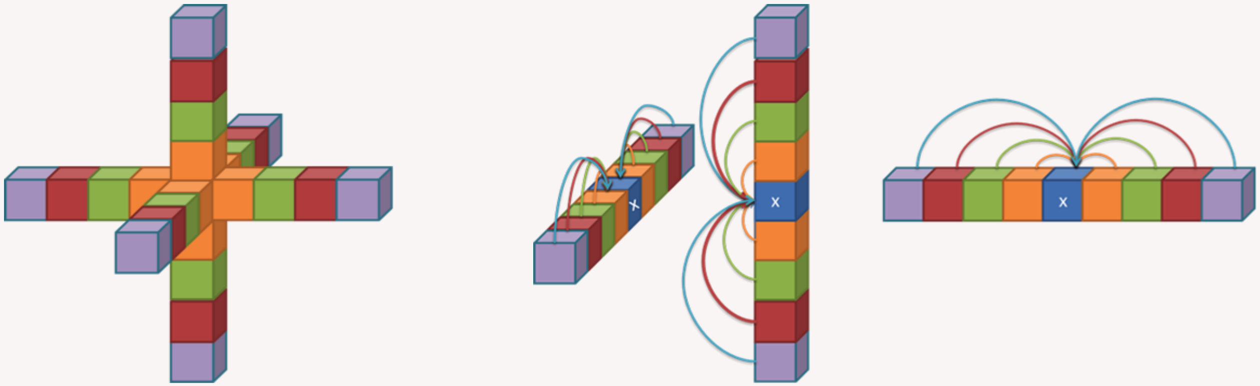


ISO-3DFD Case study

- **ISO-3DFD:** 3D Finite Difference Code with an Isotropic
- helps solving differential equations (seismic apps, wave propagation)

https://software.intel.com/en-us/articles/eight-optimizations-for-3-dimensional-finite-difference-3dfd-code-with-an-isotropic-iso

8th order stencil | **3D stencil computation**

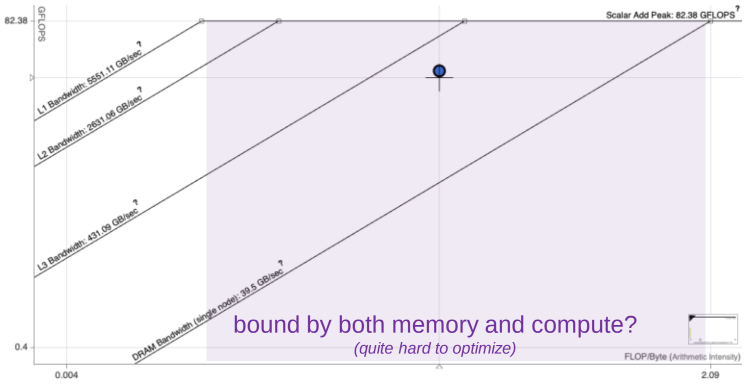


DISCLAIMER: Optimization courtesy of **Cédric Andreolli (Intel Corporation)**

Experiments ran on Intel Xeon Gold 6140 (18 cores @ 2.3GHz) 4x16GB DDR4

ISO-3DFD Case study

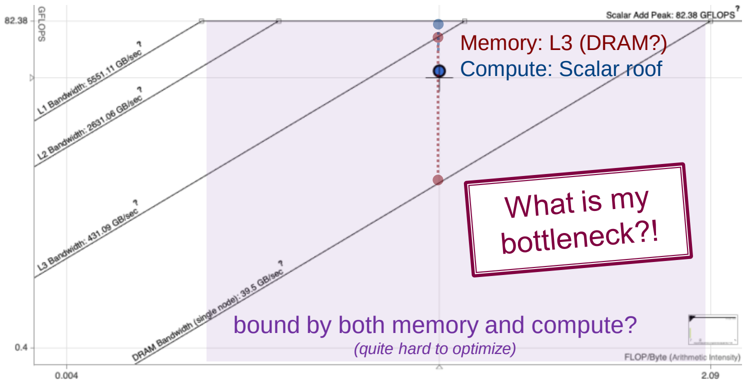
Intel Advisor CARM
(product release version)



Preliminary Outcomes: ISO-3DFD Case study

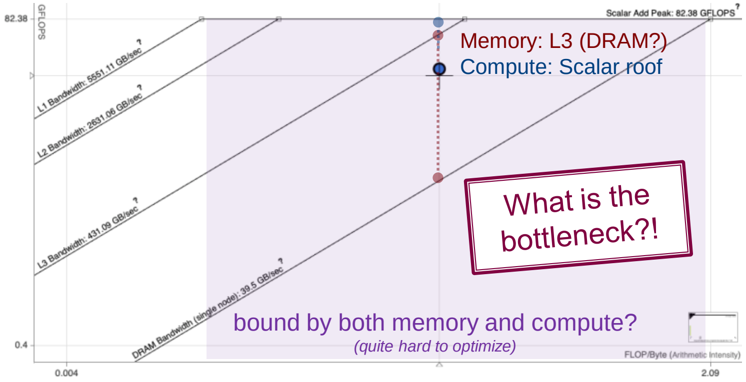
Intel Advisor CARM

(product release version)

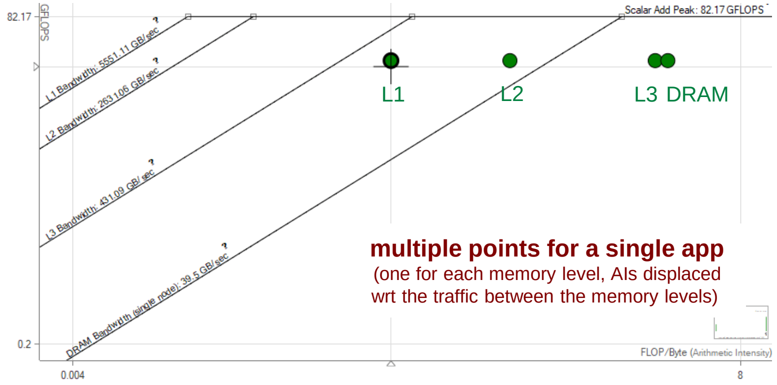


Preliminary Outcomes: ISO-3DFD Case study

Intel Advisor CARM
(product release version)



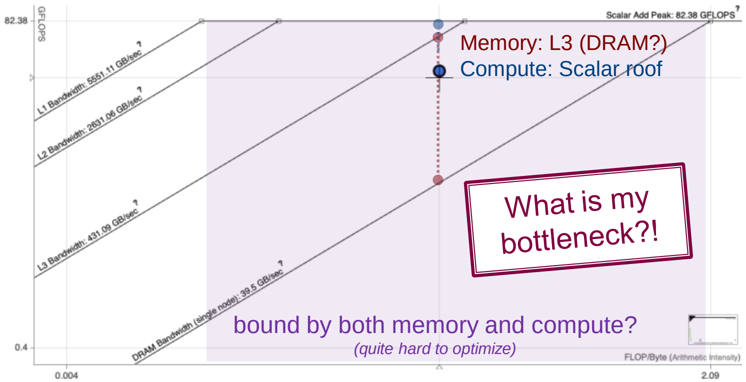
Integrated Roofline
(Advisor experimental feature)



Preliminary Outcomes: ISO-3DFD Case study

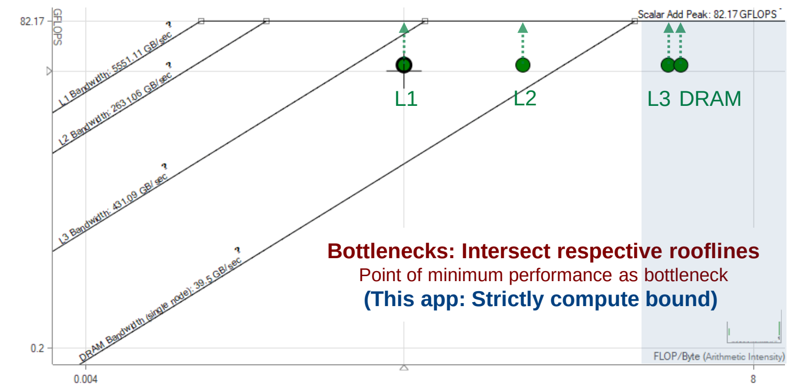
Intel Advisor CARM

(product release version)



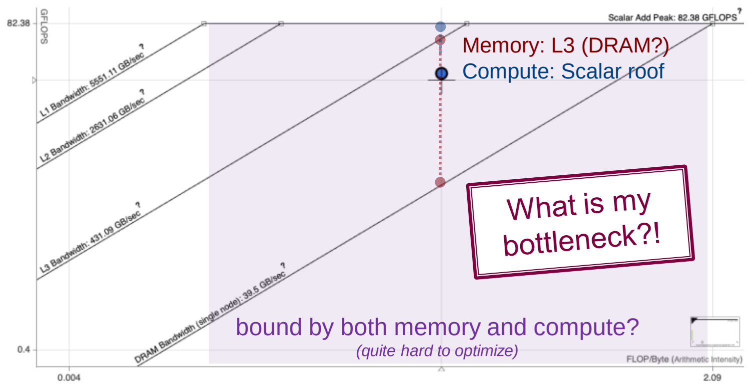
Integrated Roofline

(Advisor experimental feature)

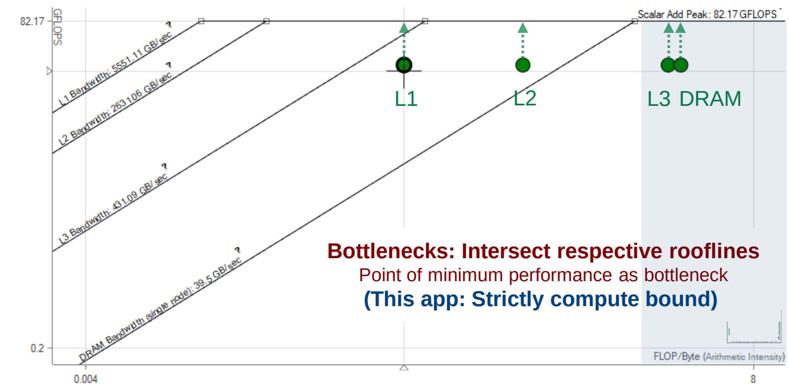


Preliminary Outcomes: ISO-3DFD Case study

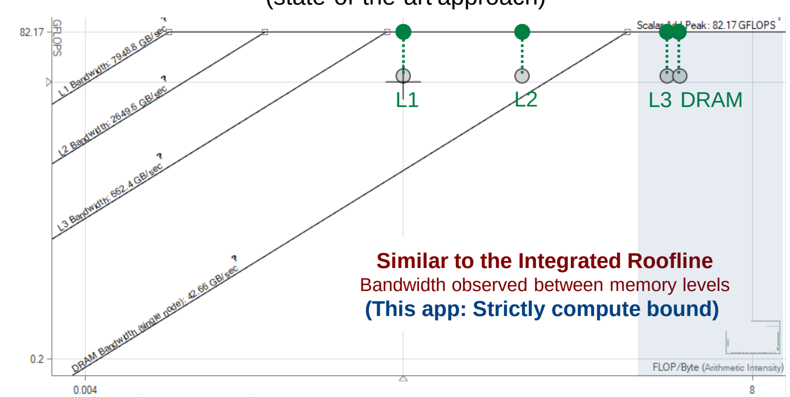
Intel Advisor CARM
(product release version)



Integrated Roofline
(Advisor experimental feature)

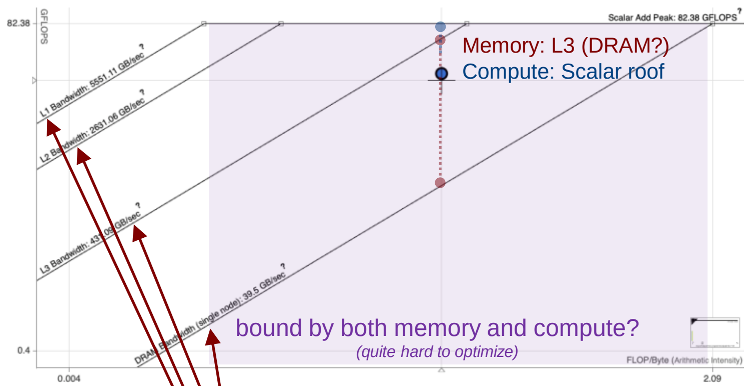


Hierarchical Roofline
(state-of-the-art approach)



Preliminary Outcomes: ISO-3DFD Case study

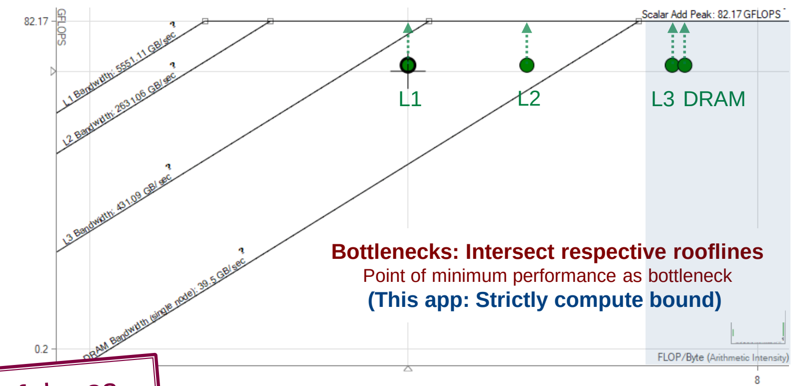
Intel Advisor CARM
(product release version)



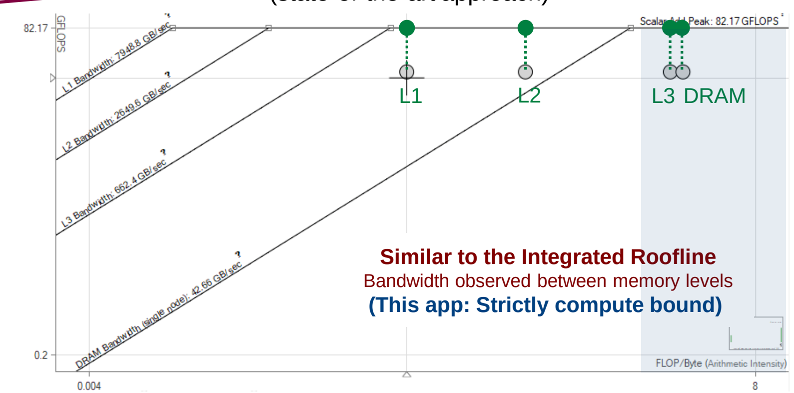
Absolute architecture maximums
(can my application exploit those?)

Can we improve insightfulness by leveraging some app-specific data?!

Integrated Roofline
(Advisor experimental feature)

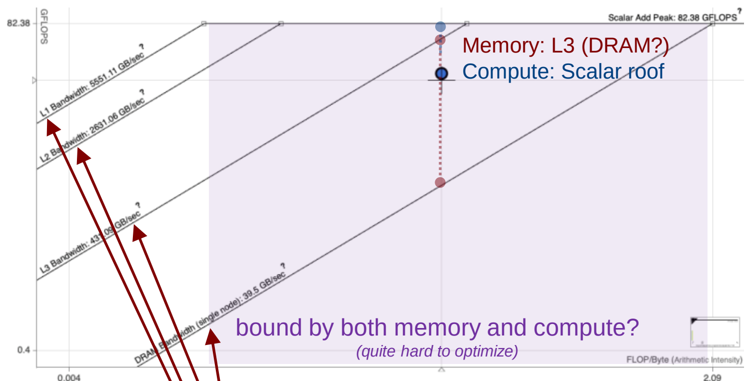


Hierarchical Roofline
(state-of-the-art approach)



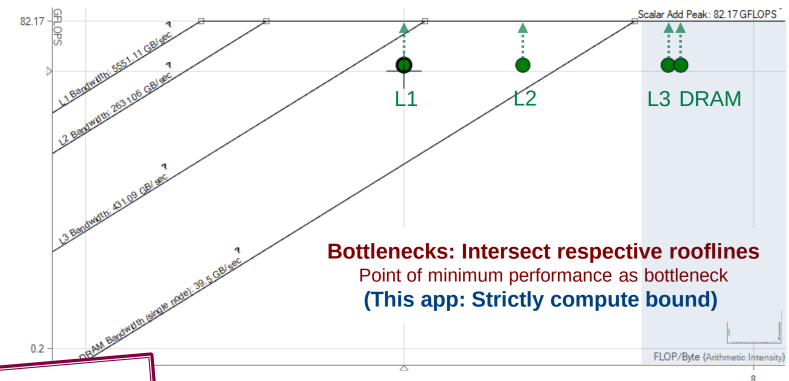
Preliminary Outcomes: ISO-3DFD Case study

Intel Advisor CARM
(product release version)



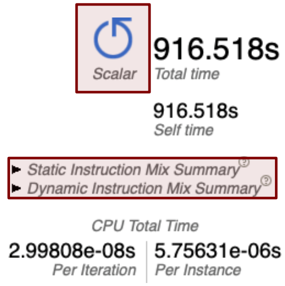
Absolute architecture maximums
(can my application exploit those?)

Integrated Roofline
(Advisor experimental feature)

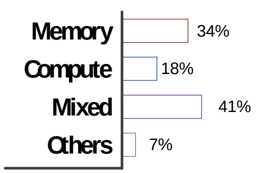


Can we improve insightfulness by leveraging some app-specific data?!

Intel Advisor



Intel SDE

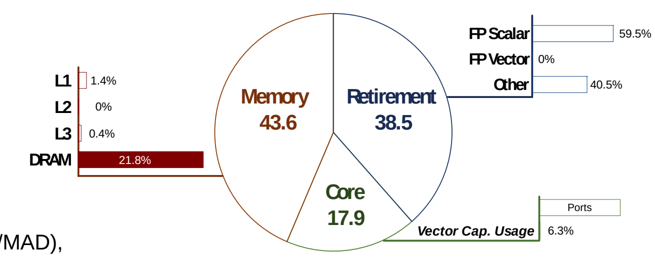


Also decoupled by:
- operation type (e.g., LD/ST, ADD/MAD),
- ISA extension (AVX512,AVX, SSE, Scalar),
- data precision (single/double) …

upon which the respective ratios are derived

Intel VTune

Micro-benchmarking
(counters, assembly)

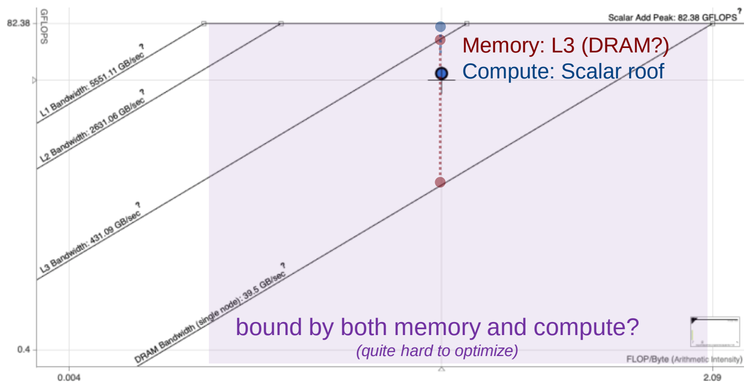


Source-code analysis

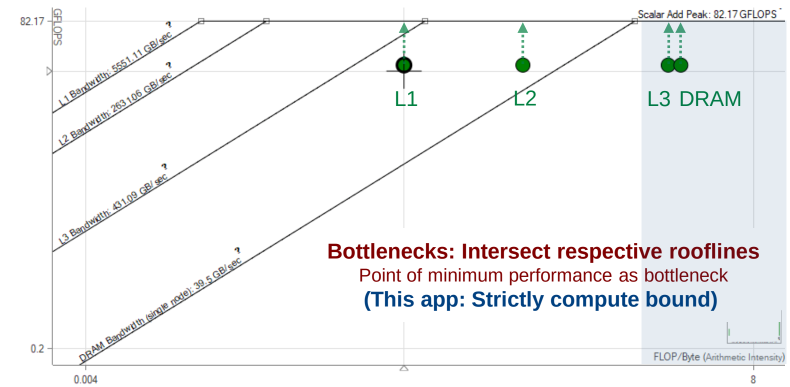
```
for(TYPE_INTEGER ix=0; ix<ixEnd; ix++) {
    value = ptr_prev[ix]*c0
        + c1 *   (FINITE_ADD(ix, 1)
            + FINITE_ADD(ix, vertical_1)
            + FINITE_ADD(ix, front_1))
        + c2 *   (FINITE_ADD(ix, 2)
            + FINITE_ADD(ix, vertical_2)
            + FINITE_ADD(ix, front_2))
        + c3 *   (FINITE_ADD(ix, 3)
            + FINITE_ADD(ix, vertical_3)
            + FINITE_ADD(ix, front_3))
        + c4 *   (FINITE_ADD(ix, 4)
            + FINITE_ADD(ix, vertical_4)
            + FINITE_ADD(ix, front_4))
#if( HALF_LENGTH == 8)
        + c5 *   (FINITE_ADD(ix, 5)
```

Preliminary Outcomes: ISO-3DFD Case study

Intel Advisor CARM
(product release version)



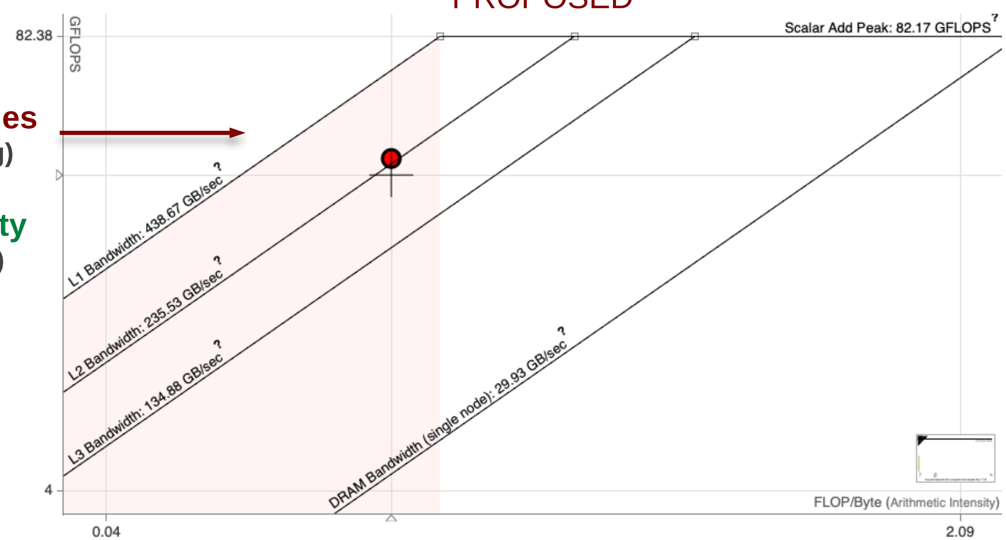
Integrated Roofline
(Advisor experimental feature)



PROPOSED

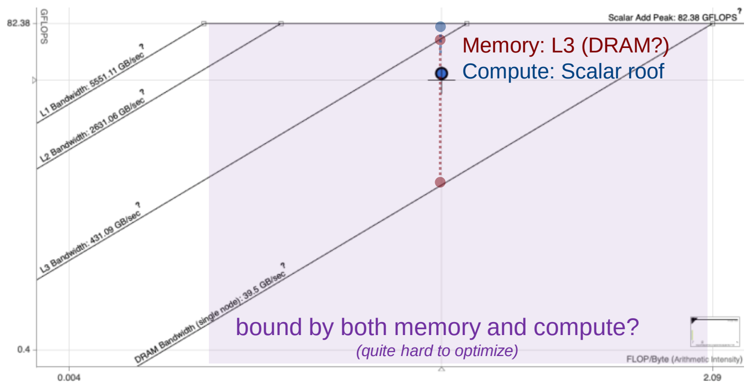
Application-driven Rooflines
(precise architecture modeling)

Preserved model simplicity
(intuitiveness, remove clutter)

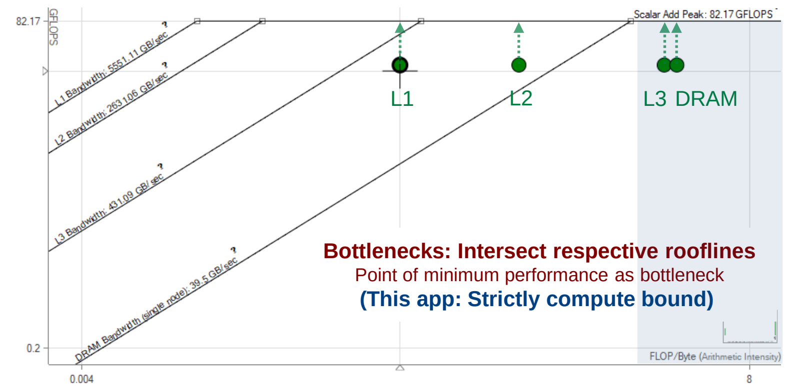


Preliminary Outcomes: ISO-3DFD Case study

Intel Advisor CARM
(product release version)



Integrated Roofline
(Advisor experimental feature)

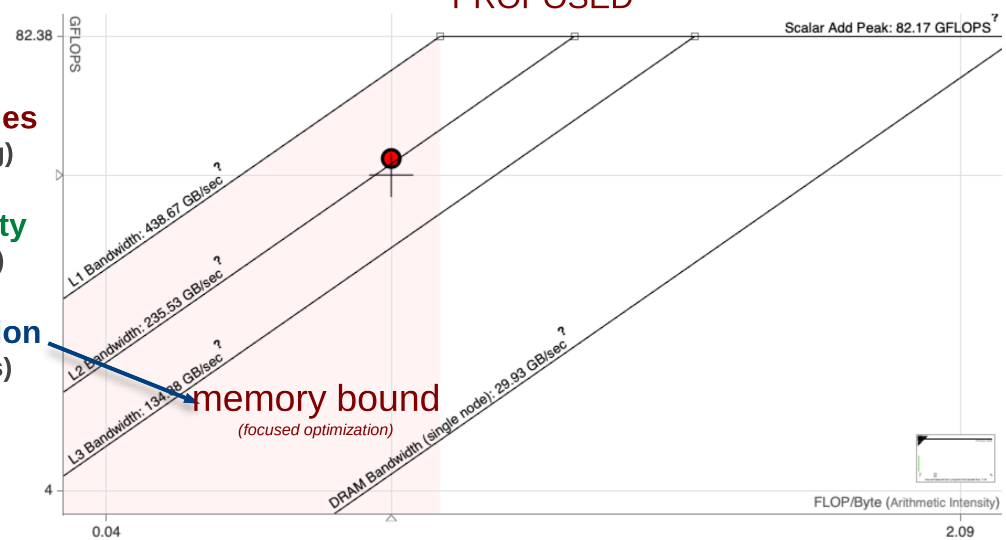


PROPOSED

Application-driven Rooflines
(precise architecture modeling)

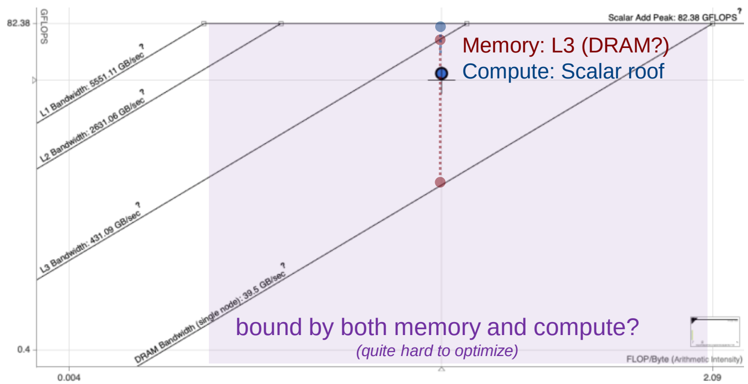
Preserved model simplicity
(intuitiveness, remove clutter)

Improved bottleneck detection
(model resembles app demands)

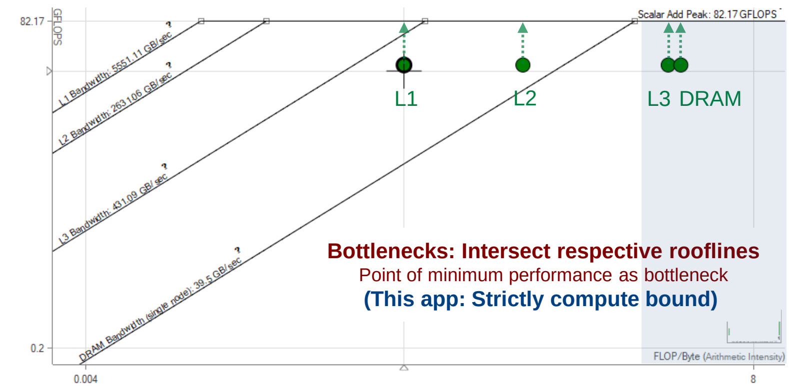


Preliminary Outcomes: ISO-3DFD Case study

Intel Advisor CARM
(product release version)



Integrated Roofline
(Advisor experimental feature)

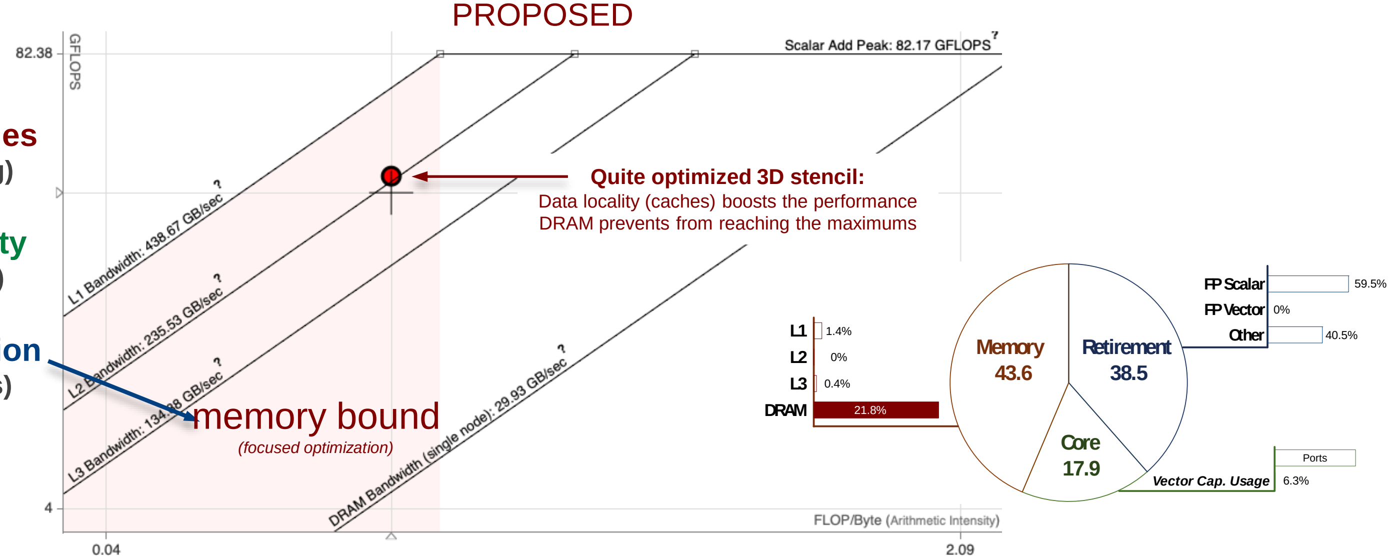


PROPOSED

Application-driven Rooflines
(precise architecture modeling)

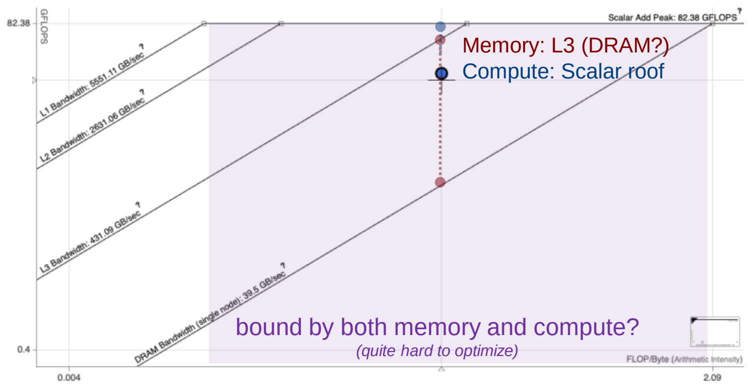
Preserved model simplicity
(intuitiveness, remove clutter)

Improved bottleneck detection
(model resembles app demands)

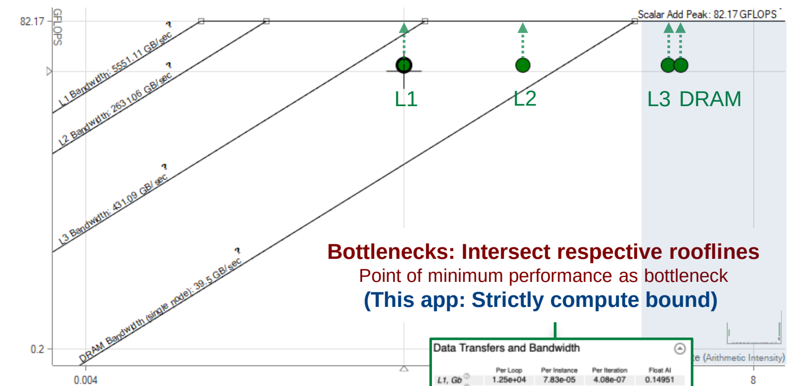


Preliminary Outcomes: ISO-3DFD Case study

Intel Advisor CARM
(product release version)



Integrated Roofline
(Advisor experimental feature)



Bottlenecks: Intersect respective rooflines
Point of minimum performance as bottleneck
(This app: Strictly compute bound)

Data Transfers and Bandwidth

	Per Loop	Per Instance	Per Iteration	Float AI
L1, Gb	1.25e+04	7.83e-05	4.08e-07	0.14951
L2, Gb	3.22e+03	2.02e-05	1.05e-07	0.579389
L3, Gb	6.15e+02	3.86e-06	2.01e-08	3.03233
DRAM, Gb	5.33e+02	3.35e-06	1.74e-08	3.50115

Self bandwidth by memory levels

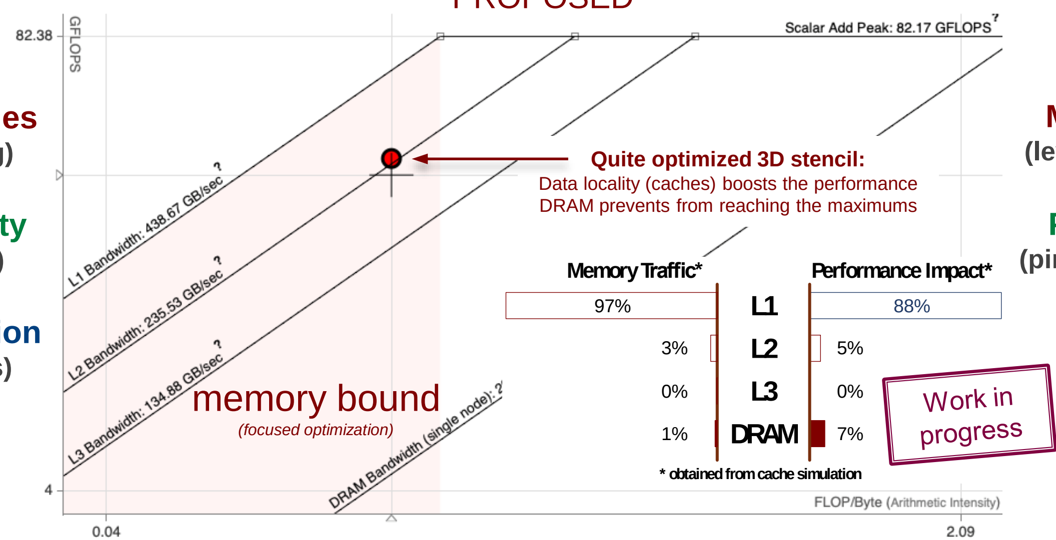
L1 Gb/s	244.138
L2 Gb/s	62.999
L3 Gb/s	12.0373
DRAM Gb/s	10.4254

PROPOSED

Application-driven Rooflines
(precise architecture modeling)

Preserved model simplicity
(intuitiveness, remove clutter)

Improved bottleneck detection
(model resembles app demands)



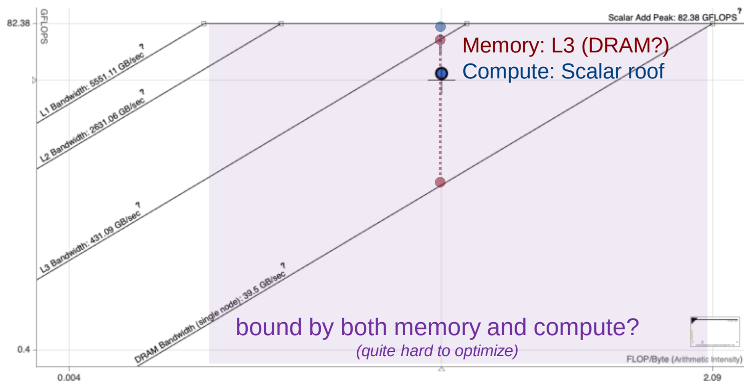
Memory traffic shares
(leverage other Advisor data)

Performance impact
(pinpointing the bottlenecks)

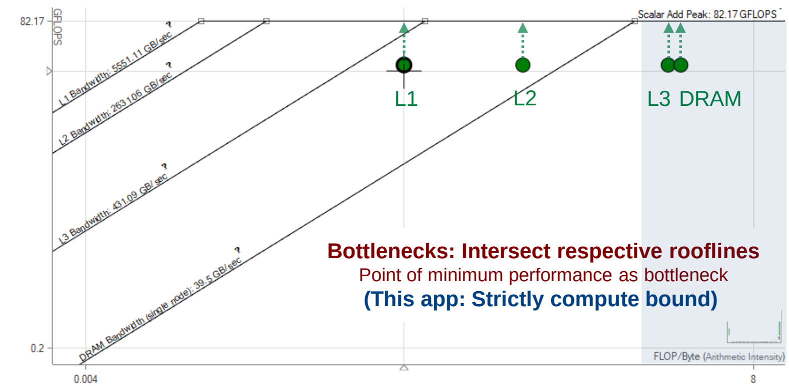
Preliminary Outcomes: ISO-3DFD Case study

inesc id lisboa | TÉCNICO LISBOA

Intel Advisor CARM
(product release version)



Integrated Roofline
(Advisor experimental feature)



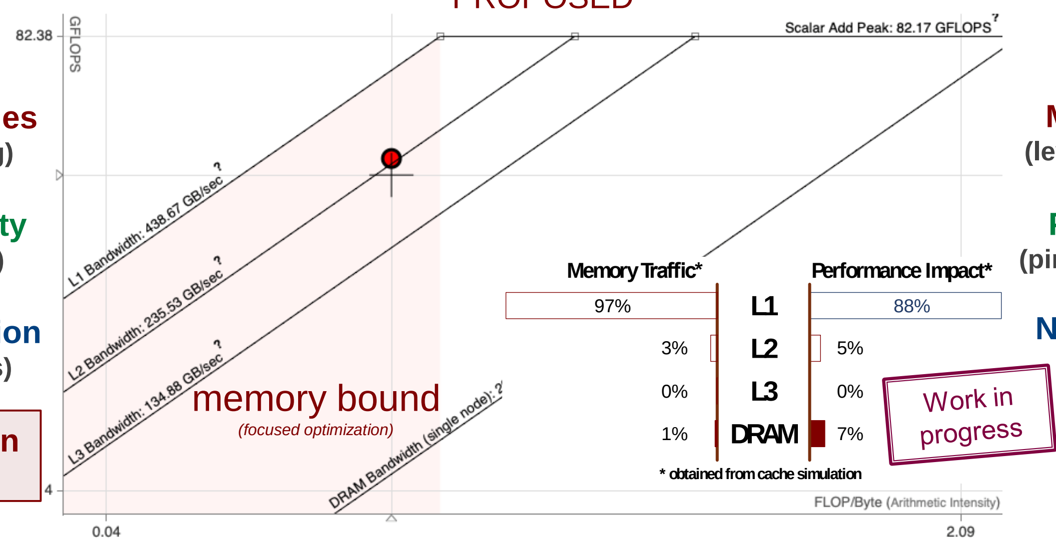
PROPOSED

Application-driven Rooflines
(precise architecture modeling)

Preserved model simplicity
(intuitiveness, remove clutter)

Improved bottleneck detection
(model resembles app demands)

Consistent characterization
(eases code vectorization)



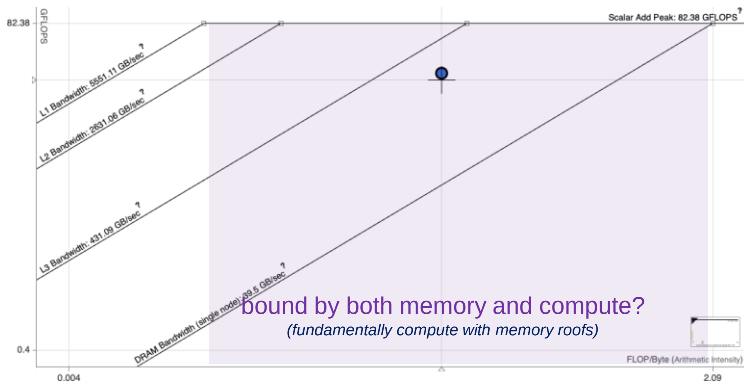
Memory traffic shares
(leverage other Advisor data)

Performance impact
(pinpointing the bottlenecks)

New set of visual aids
(to drive optimization)

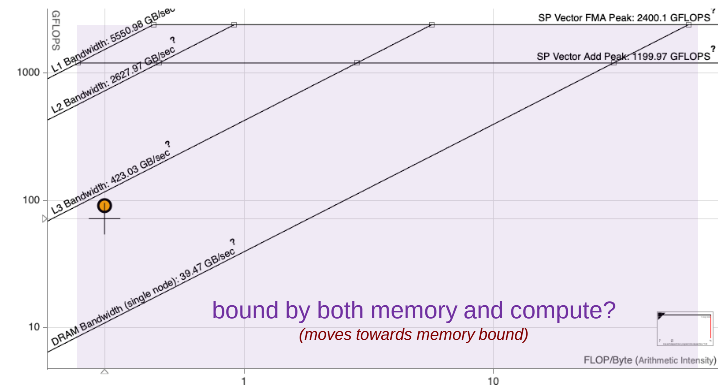
Preliminary Outcomes:
ISO-3DFD Case study

Intel Advisor CARM
(product release version)



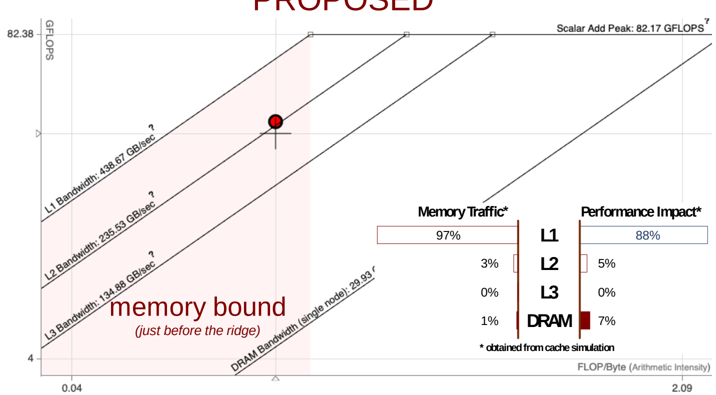
Scalar

Intel Advisor CARM
(product release version)

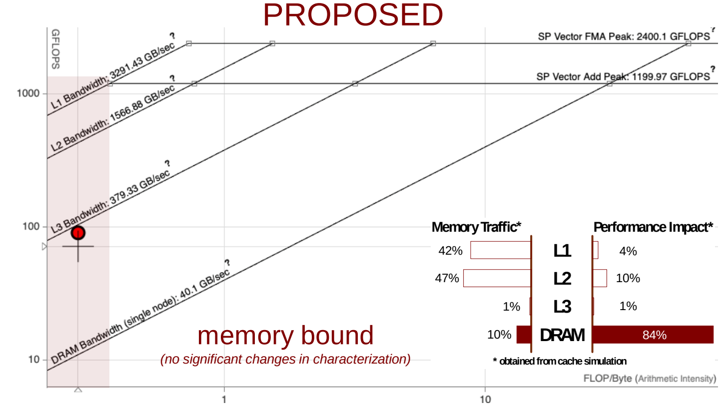


AVX-512

PROPOSED

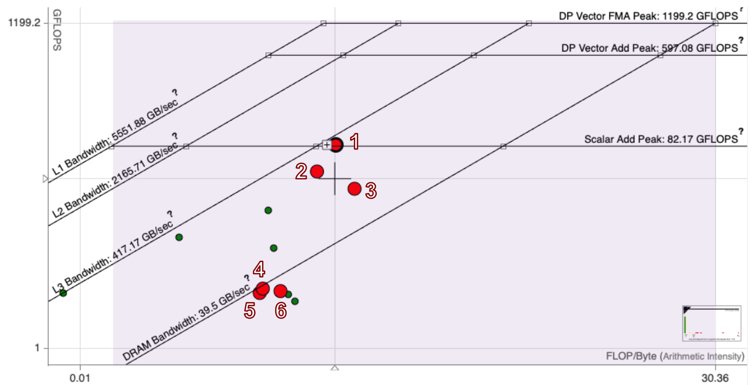


PROPOSED



Proxy-applications from Exascale Computing Project

Intel Advisor CARM
(product release version)



sw4lite (LLNL, US)

Proxy version of SW4 (3-D seismic modeling)

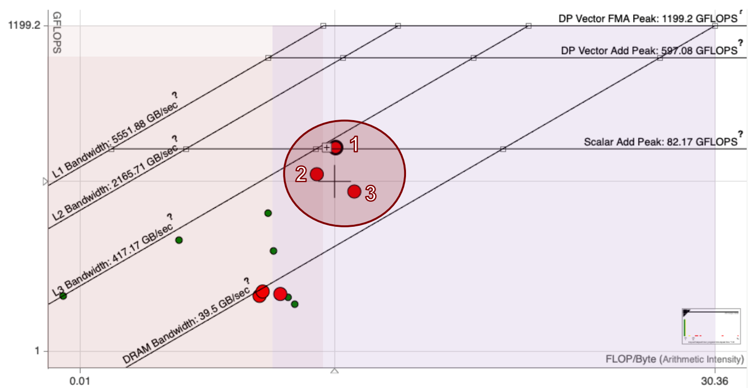
- Again bunch of stencils ….
- 6 main hotspots (loops)

Intel Advisor CARM:

- **All loops are bound by both (mem and comp)**
- **Loops 1 and 2:** Mainly limited by L3
- **Loop 3:** Between DRAM and L3 (some locality)
- **Loops 4, 5 and 6:** DRAM bound

Proxy-applications from Exascale Computing Project

Intel Advisor CARM
(product release version)



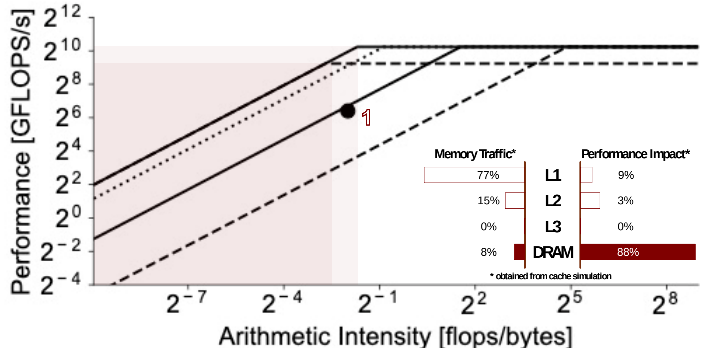
sw4lite (LLNL, US)

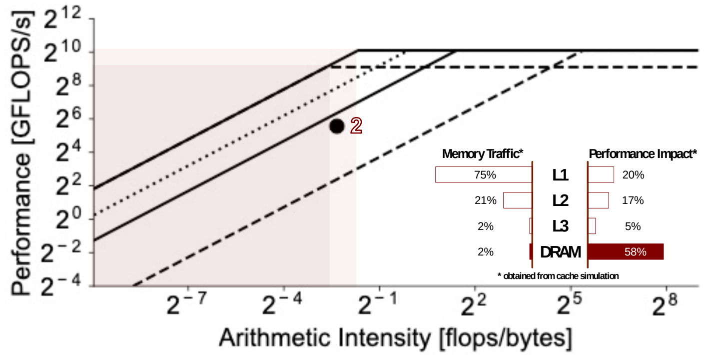
Proxy version of SW4 (3-D seismic modeling)

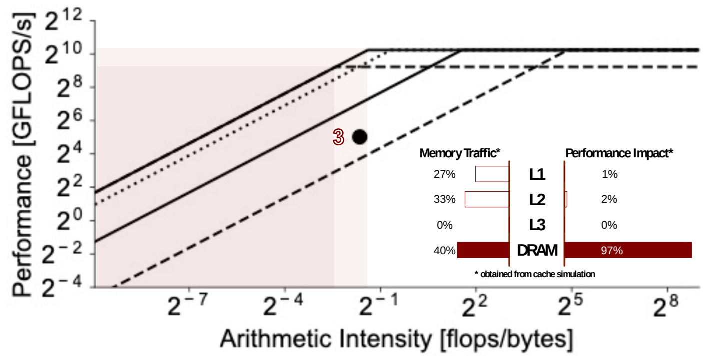
- Again bunch of stencils ….
- 6 main hotspots (loops)

Improved interpretation methodology in action:

- **Memory traffic shares**: Additional performance insights (explains the dot position)
- **Performance impact**: Improved optimization hints (decouples the bottlenecks by their importance)

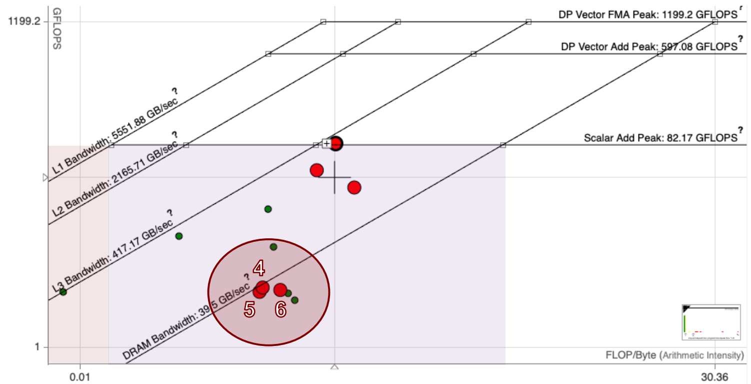






Proxy-applications from Exascale Computing Project

Intel Advisor CARM
(product release version)



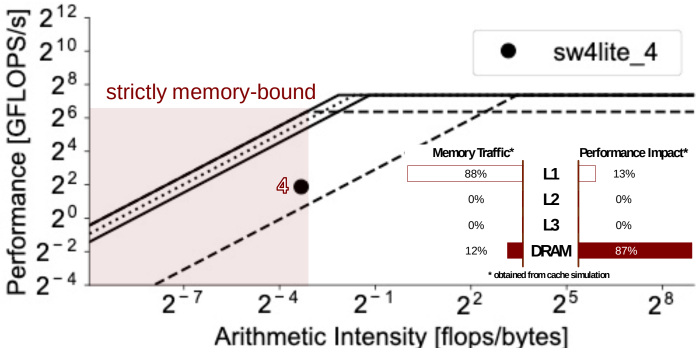
sw4lite (LLNL, US)

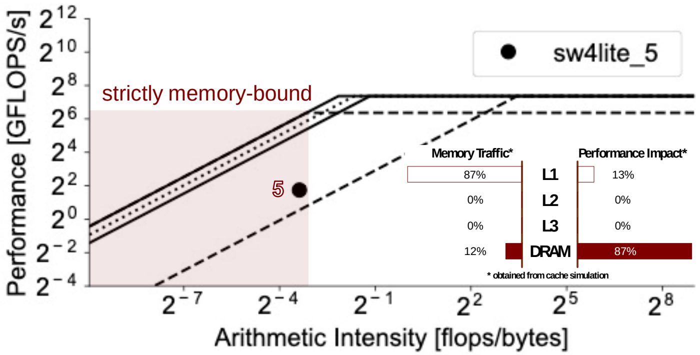
bare bone version of SW4 (3-D seismic modeling)

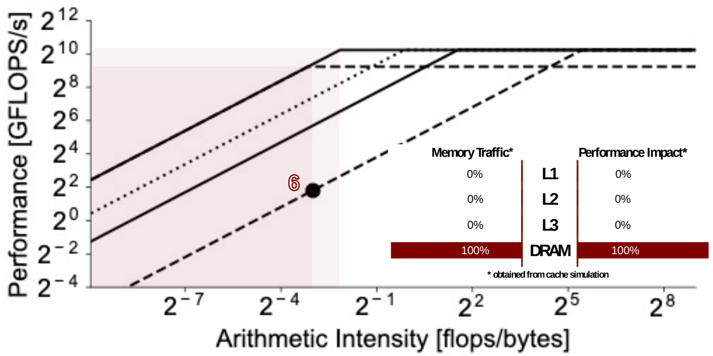
- Again bunch of stencils ….
- 6 main hotspots (loops)

Improved interpretation methodology in action:

- **Memory traffic shares**: Additional performance insights (explains the dot position)
- **Performance impact**: Improved optimization hints (decouples the bottlenecks by their importance)

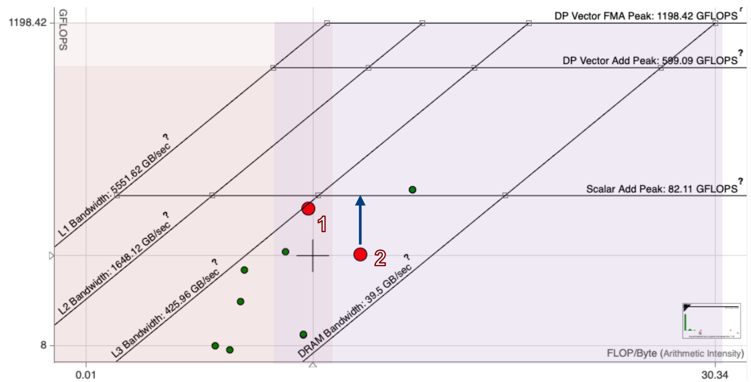


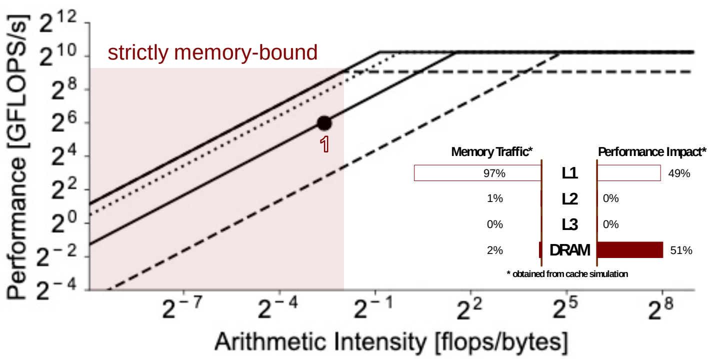




Proxy-applications from Exascale Computing Project

Intel Advisor CARM
(product release version)

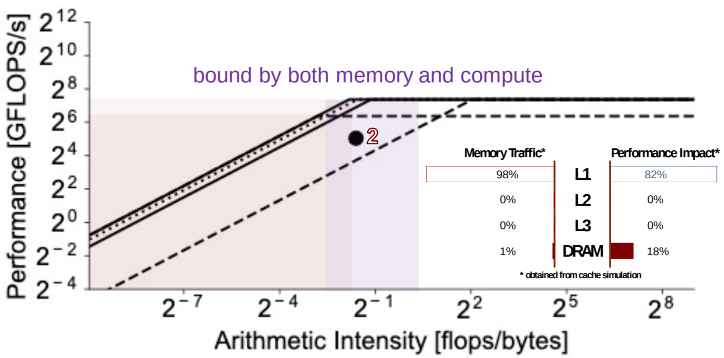




ExaMiniMD (SNL)
Molecular Dynamics

- 2 main hotspots (loops)
- Loop 1 (memory bound), Loop 2 (comp/mem)

Improved interpretation methodology in action:
- **Memory traffic shares**: Additional performance insights (explains the dot position)
- **Performance impact**: Improved optimization hints (decouples the bottlenecks by their importance)



Bayesian K2 Score

How to detect which genes influence traits or diseases?

Bayesian K2 Score

How to detect which genes influence traits or diseases?

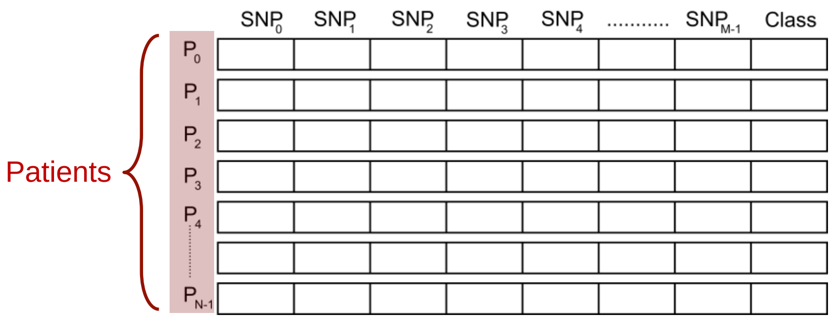
Bayesian K2 Score

High-Order Epistasis

- Relates genetic markers that are most likely to influence diseases.
- Calculates score for a combination of K genetic markers (order).
- Highest score corresponds to SNP combination that is most likely to influence the trait or disease.

Bayesian K2 Score

How to detect which genes influence traits or diseases?



Bayesian K2 Score

High-Order Epistasis

- Relates genetic markers that are most likely to influence traits or diseases.
- Calculates score for a combination of K genetic markers (order).
- Highest score corresponds to SNP combination that is most likely to influence the trait or disease.

Bayesian K2 Score

How to detect which genes influence traits or diseases?

Genetic Markers

Patients

	SNP_0	SNP_1	SNP_2	SNP_3	SNP_4	....	SNP_{M-1}	Class
P_0								
P_1								
P_2								
P_3								
P_4								
P_{N-1}								

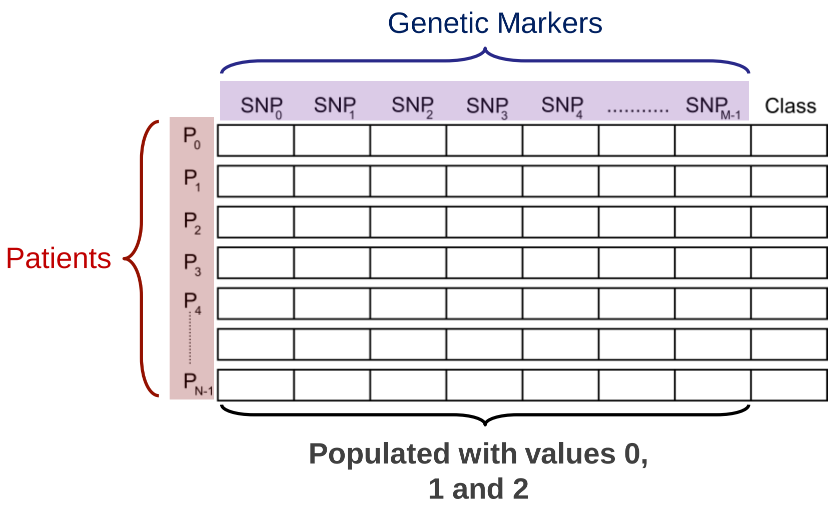
Bayesian K2 Score

High-Order Epistasis

- Relates genetic markers that are most likely to influence traits or diseases.
- Calculates score for a combination of K genetic markers (order).
- Highest score corresponds to SNP combination that is most likely to influence the trait or disease.

Bayesian K2 Score

How to detect which genes influence traits or diseases?



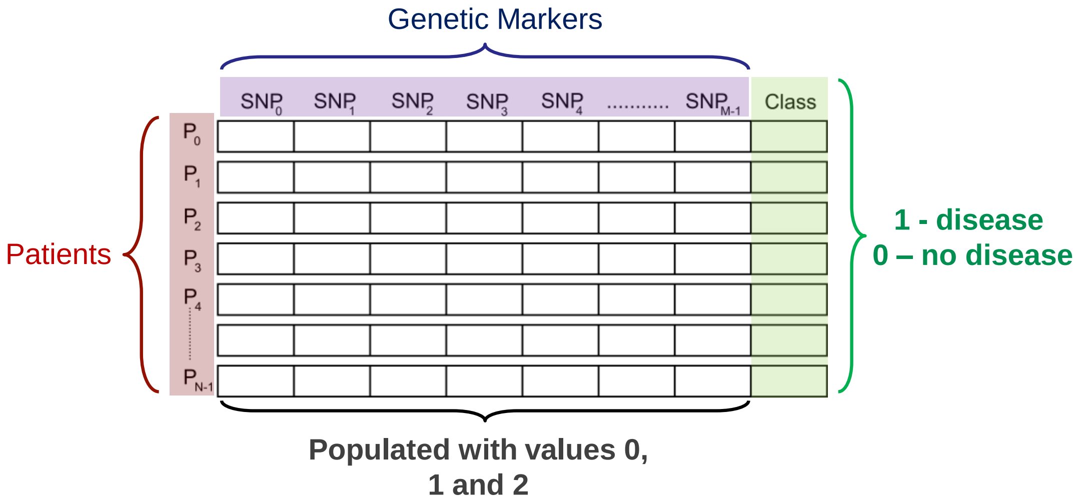
Bayesian K2 Score

High-Order Epistasis

- Relates genetic markers that are most likely to influence traits or diseases.
- Calculates score for a combination of K genetic markers (order).
- Highest score corresponds to SNP combination that is most likely to influence the trait or disease.

Bayesian K2 Score

How to detect which genes influence traits or diseases?



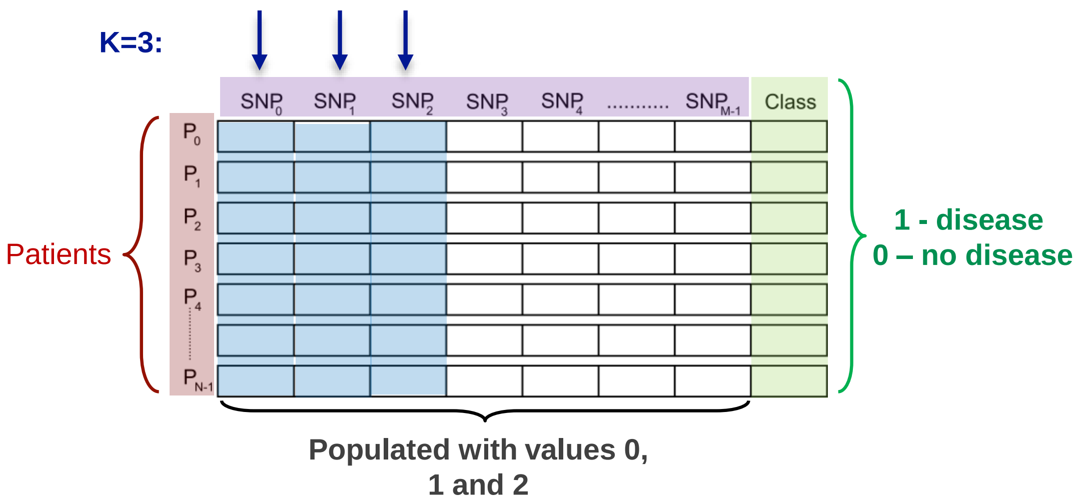
Bayesian K2 Score

High-Order Epistasis

- Relates genetic markers that are most likely to influence traits or diseases.
- Calculates score for a combination of K genetic markers (order).
- Highest score corresponds to SNP combination that is most likely to influence the trait or disease.

Bayesian K2 Score

How to detect which genes influence traits or diseases?



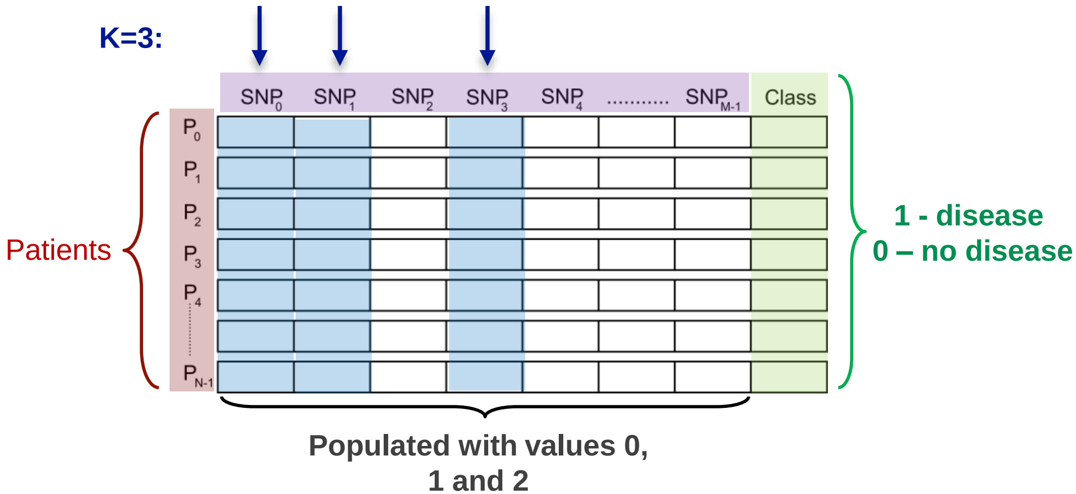
Bayesian K2 Score

High-Order Epistasis

- Relates genetic markers that are most likely to influence traits or diseases.
- Calculates score for a combination of K genetic markers (order).
- Highest score corresponds to SNP combination that is most likely to influence the trait or disease.

Bayesian K2 Score

How to detect which genes influence traits or diseases?



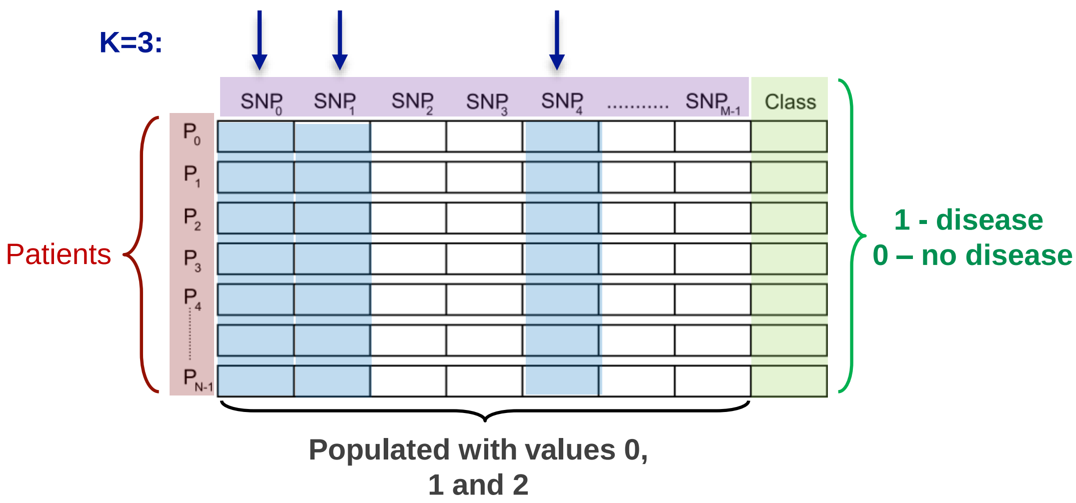
Bayesian K2 Score

High-Order Epistasis

- Relates genetic markers that are most likely to influence traits or diseases.
- Calculates score for a combination of K genetic markers (order).
- Highest score corresponds to SNP combination that is most likely to influence the trait or disease.

Bayesian K2 Score

How to detect which genes influence traits or diseases?



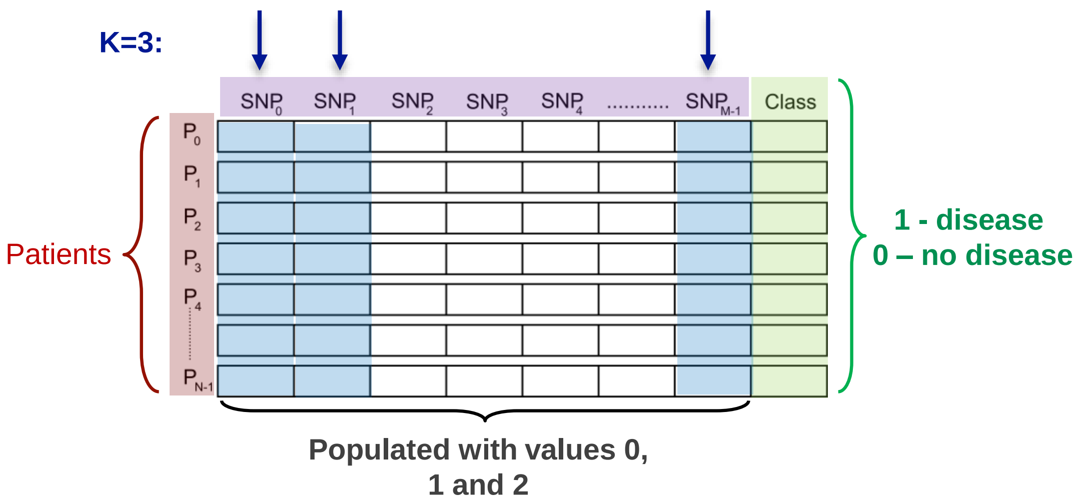
Bayesian K2 Score

High-Order Epistasis

- Relates genetic markers that are most likely to influence traits or diseases.
- Calculates score for a combination of K genetic markers (order).
- Highest score corresponds to SNP combination that is most likely to influence the trait or disease.

Bayesian K2 Score

How to detect which genes influence traits or diseases?

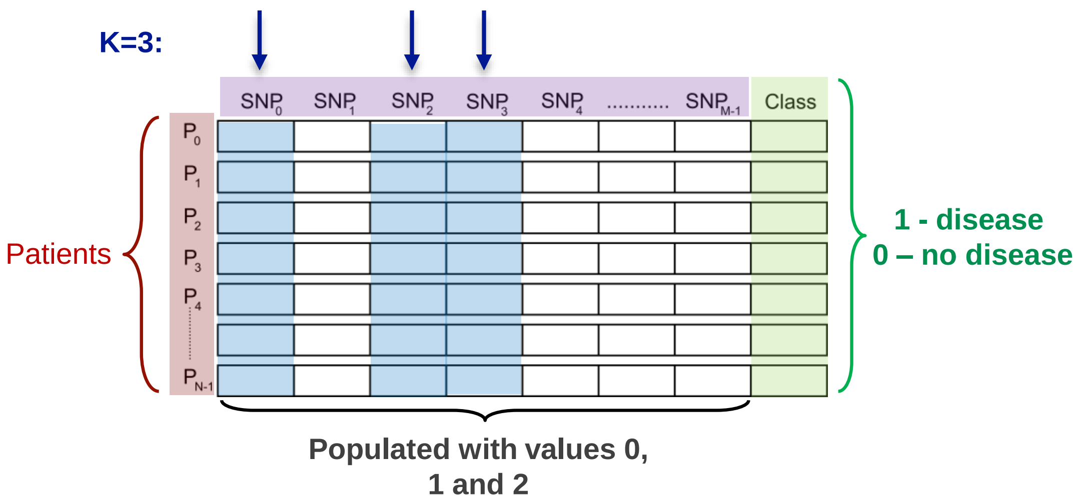


Bayesian K2 Score
High-Order Epistasis

- Relates genetic markers that are most likely to influence traits or diseases.
- Calculates score for a combination of K genetic markers (order).
- Highest score corresponds to SNP combination that is most likely to influence the trait or disease.

Bayesian K2 Score

How to detect which genes influence traits or diseases?



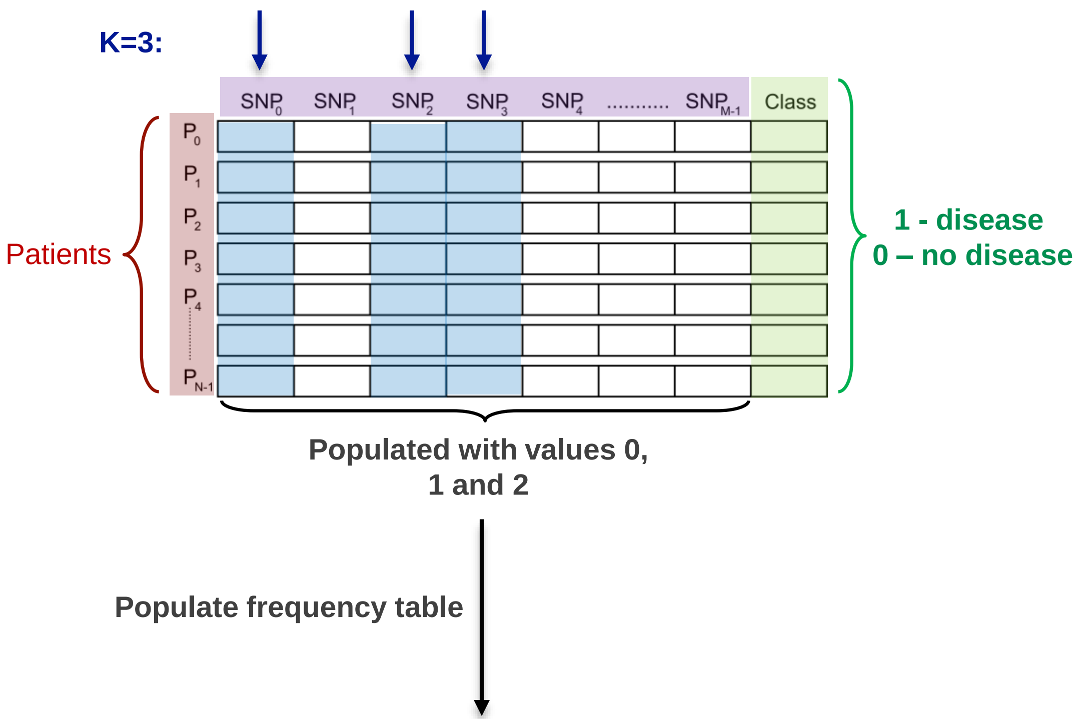
Bayesian K2 Score

High-Order Epistasis

- Relates genetic markers that are most likely to influence traits or diseases.
- Calculates score for a combination of K genetic markers (order).
- Highest score corresponds to SNP combination that is most likely to influence the trait or disease.

Bayesian K2 Score

How to detect which genes influence traits or diseases?



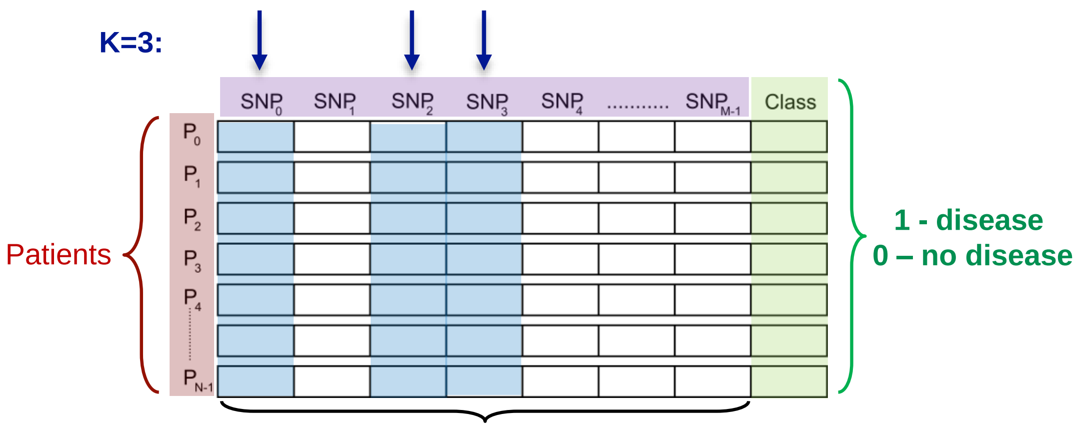
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Class = 0																X											
Class = 1																Y											

Bayesian K2 Score
High-Order Epistasis

- Relates genetic markers that are most likely to influence traits or diseases.
- Calculates score for a combination of K genetic markers (order).
- Highest score corresponds to SNP combination that is most likely to influence the trait or disease.

Bayesian K2 Score

How to detect which genes influence traits or diseases?



Populated with values 0, 1 and 2

Populate frequency table

#Columns = 3^k

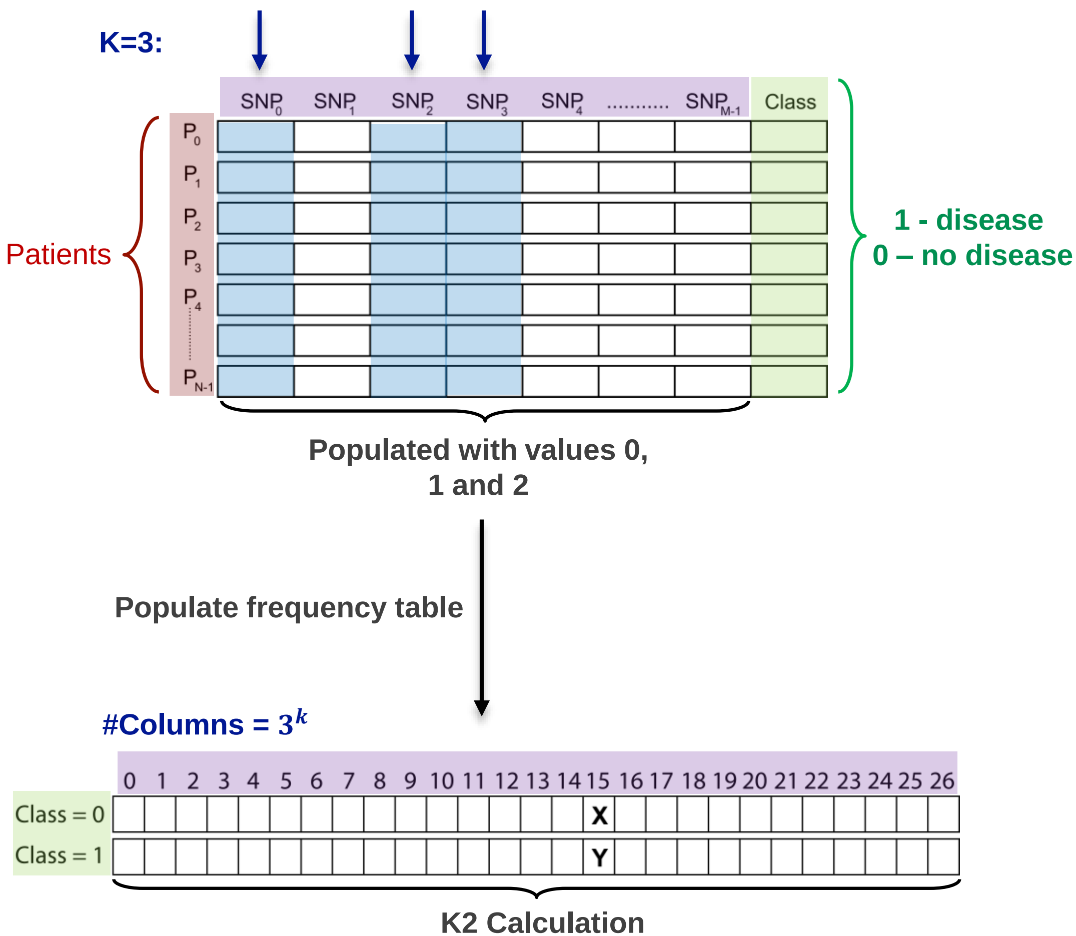
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Class = 0																X											
Class = 1																Y											

Bayesian K2 Score
High-Order Epistasis

- Relates genetic markers that are most likely to influence traits or diseases.
- Calculates score for a combination of K genetic markers (order).
- Highest score corresponds to SNP combination that is most likely to influence the trait or disease.

Bayesian K2 Score

How to detect which genes influence traits or diseases?



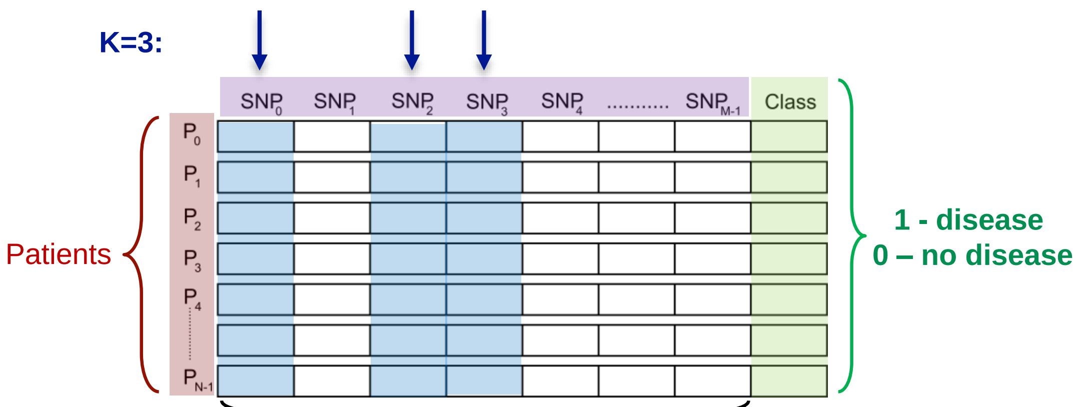
Bayesian K2 Score

High-Order Epistasis

- Relates genetic markers that are most likely to influence traits or diseases.
- Calculates score for a combination of K genetic markers (order).
- Highest score corresponds to SNP combination that is most likely to influence the trait or disease.

Bayesian K2 Score

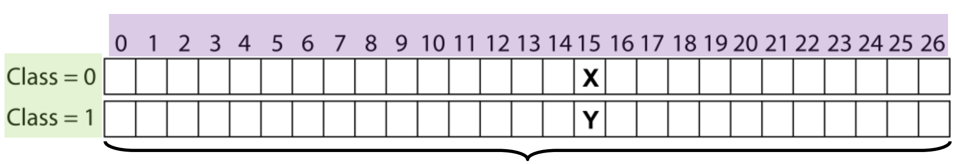
How to detect which genes influence traits or diseases?



Populated with values 0, 1 and 2

Populate frequency table

#Columns = 3^k



K2 Calculation

Bayesian K2 Score

High-Order Epistasis

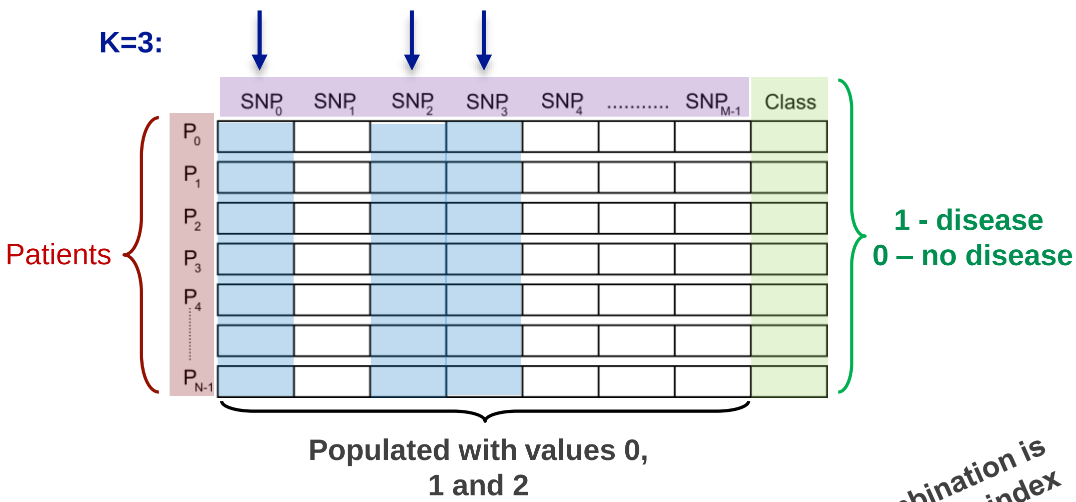
- Relates genetic markers that are most likely to influence traits or diseases.
- Calculates score for a combination of K genetic markers (order).
- Highest score corresponds to SNP combination that is most likely to influence the trait or disease.

K2 Score:

$$\boldsymbol{K2} = \sum_{j=1}^{X+Y+1} \log\,(j) + \sum_{j=1}^{X} \boldsymbol{log}\,(j) + \sum_{j=1}^{Y} \boldsymbol{log}\,(j)$$

Bayesian K2 Score

How to detect which genes influence traits or diseases?



Bayesian K2 Score
High-Order Epistasis

- Relates genetic markers that are most likely to influence traits or diseases.
- Calculates score for a combination of K genetic markers (order).
- Highest score corresponds to SNP combination that is most likely to influence the trait or disease.

SNP_2	0									1									2								
SNP_1	0			1			2			0			1			2			0			1			2		
SNP_0	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2

Each combination is converted to a index

Populate frequency table

#Columns = 3^k

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Class = 0																X											
Class = 1																Y											

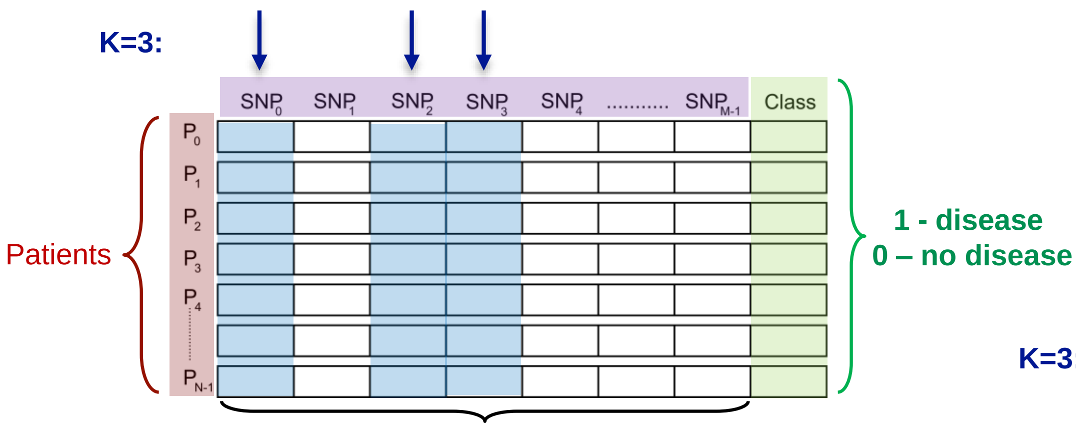
K2 Calculation

K2 Score:

$$K2 = \sum_{j=1}^{X+Y+1} \log\,(j) + \sum_{j=1}^{X} \log\,(j) + \sum_{j=1}^{Y} \log\,(j)$$

Bayesian K2 Score

How to detect which genes influence traits or diseases?



Populated with values 0, 1 and 2

Populate frequency table

Bayesian K2 Score

High-Order Epistasis

- Relates genetic markers that are most likely to influence traits or diseases.
- Calculates score for a combination of K genetic markers (order).
- Highest score corresponds to SNP combination that is most likely to influence the trait or disease.

K=3:

<table>
<tr><td>SNP2</td><td colspan="9">0</td><td colspan="9">1</td><td colspan="9">2</td></tr>
<tr><td>SNP1</td><td colspan="3">0</td><td colspan="3">1</td><td colspan="3">2</td><td colspan="3">0</td><td colspan="3">1</td><td colspan="3">2</td><td colspan="3">0</td><td colspan="3">1</td><td colspan="3">2</td></tr>
<tr><td>SNP0</td><td>0</td><td>1</td><td>2</td><td>0</td><td>1</td><td>2</td><td>0</td><td>1</td><td>2</td><td>0</td><td>1</td><td>2</td><td>0</td><td>1</td><td>2</td><td>0</td><td>1</td><td>2</td><td>0</td><td>1</td><td>2</td><td>0</td><td>1</td><td>2</td><td>0</td><td>1</td><td>2</td></tr>
</table>

Each combination is converted to a index

Index Calculation:

$$IDX = (1) \text{ x } 3$$

#Columns = 3^k

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Class = 0																X											
Class = 1																Y											

K2 Calculation

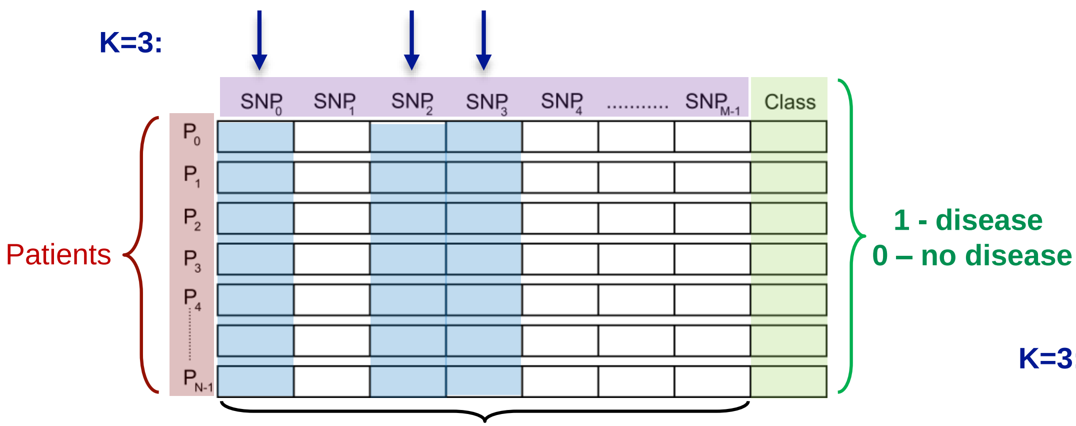
K2 Score:

$$K2 = \sum_{j=1}^{X+Y+1} \log(j) + \sum_{j=1}^{X} \log(j) + \sum_{j=1}^{Y} \log(j)$$

Bayesian K2 Score

How to detect which genes influence traits or diseases?

K=3:



Populated with values 0, 1 and 2

Populate frequency table

Bayesian K2 Score

High-Order Epistasis

- Relates genetic markers that are most likely to influence traits or diseases.
- Calculates score for a combination of K genetic markers (order).
- Highest score corresponds to SNP combination that is most likely to influence the trait or disease.

K=3:

SNP_2	0									1									2								
SNP_1	0			1			2			0			1			2			0			1			2		
SNP_0	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2

Each combination is converted to a index

Index Calculation:

$$IDX = \big(((1)\ \mathrm{x}\ 3) + 2\big)\ \mathrm{x}\ 3$$

#Columns = 3^k

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Class = 0																X											
Class = 1																Y											

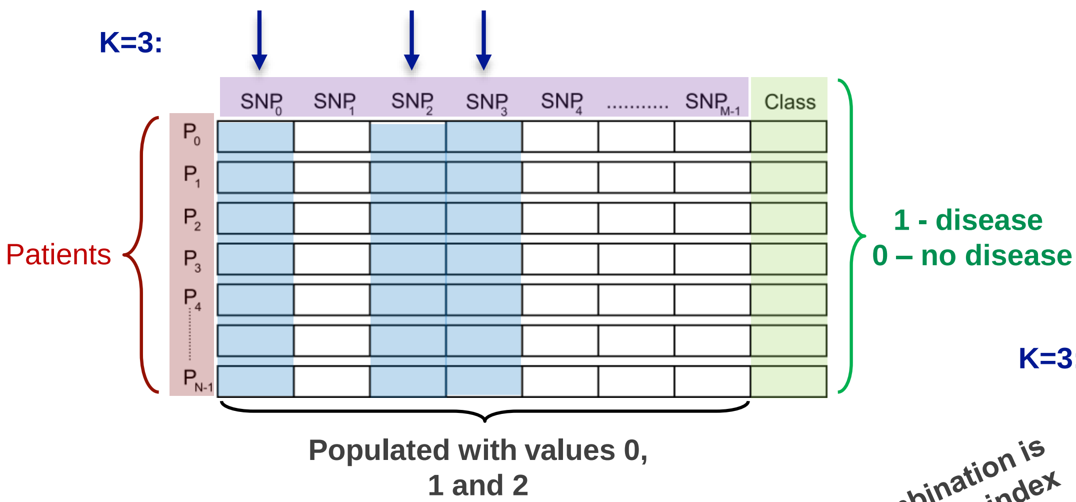
K2 Calculation

K2 Score:

$$K2 = \sum_{j=1}^{X+Y+1} \log(j) + \sum_{j=1}^{X} \log(j) + \sum_{j=1}^{Y} \log(j)$$

Bayesian K2 Score

How to detect which genes influence traits or diseases?



Bayesian K2 Score

High-Order Epistasis

- Relates genetic markers that are most likely to influence traits or diseases.
- Calculates score for a combination of K genetic markers (order).
- Highest score corresponds to SNP combination that is most likely to influence the trait or disease.

K=3:

SNP_2	0									1									2								
SNP_1	0			1			2			0			1			2			0			1			2		
SNP_0	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2

Index Calculation:

$$IDX = \left(\left((1)\ x\ 3 + 2\right)\ x\ 3 + 0\right)$$

Each combination is converted to a index

Populate frequency table

#Columns = 3^k

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Class = 0																X											
Class = 1																Y											

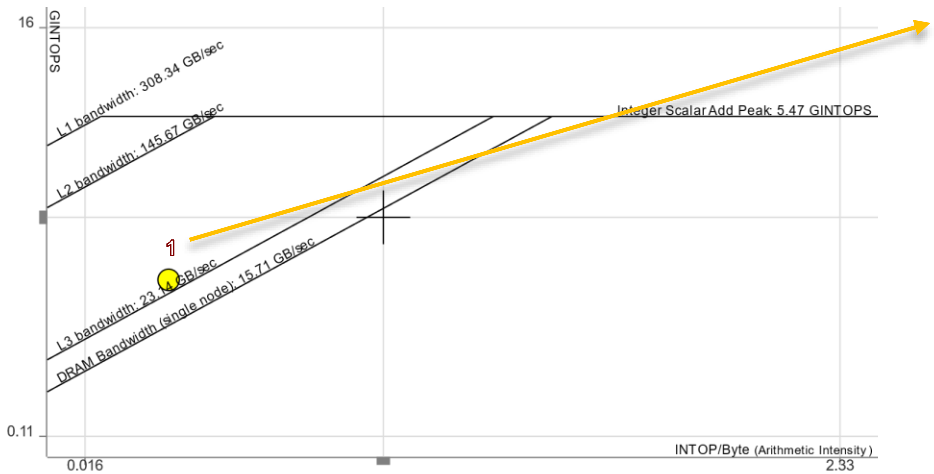
K2 Calculation

K2 Score:

$$K2 = \sum_{j=1}^{X+Y+1} \log\ (j) + \sum_{j=1}^{X} log\ (j) + \sum_{j=1}^{Y} log\ (j)$$

Bayesian K2 Score

Original Implementation
Intel Advisor CARM

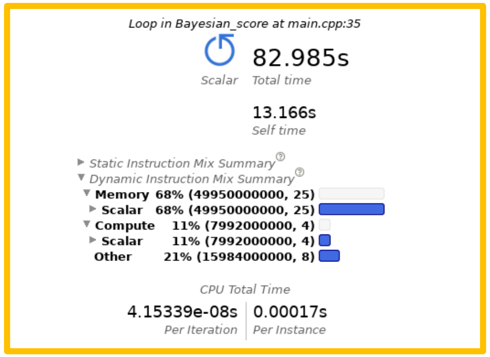


Bayesian K2 Score
High-Order Epistasis
- 1 main hotspot **Intel Advisor CARM:** - **Loops is bound by both (mem and comp)** - **Loop 1:** Mainly limited by L3

Index Calculation + Frequency Table

```
for(i=0;i<SNPdata.samplesize;i++) {
    index = 0;
    cont = 1;
    for(j=0;j<k;j++) {
        if(SNPdata.data[i][selectedSNPSet[j]] == 3) {
            cont = 0;
            break;
        }
        else
            index = index + SNPdata.data[i][selectedSNPSet[j]]*(int)pow(3.0,(k-1-j));
    }
    if(cont) {
        observedValues[SNPdata.data[i][SNPdata.data_col-1]][index]++;
        colSumTable[index]++;
    }
}
```

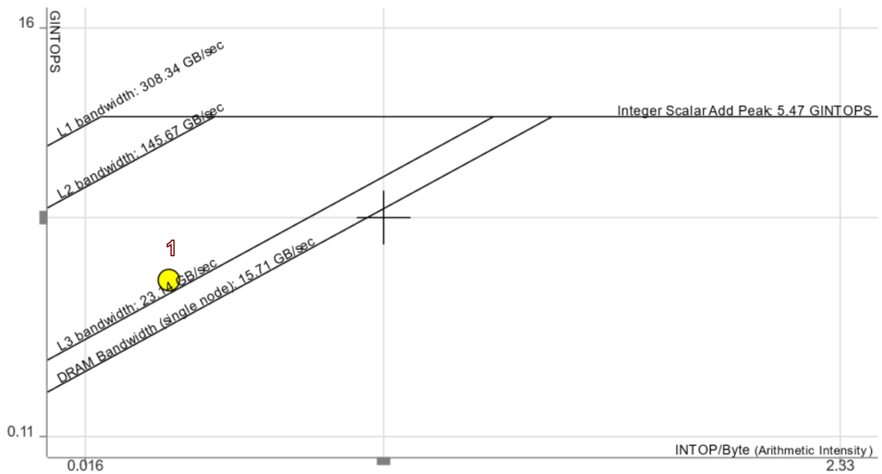
Advisor Instruction Mix

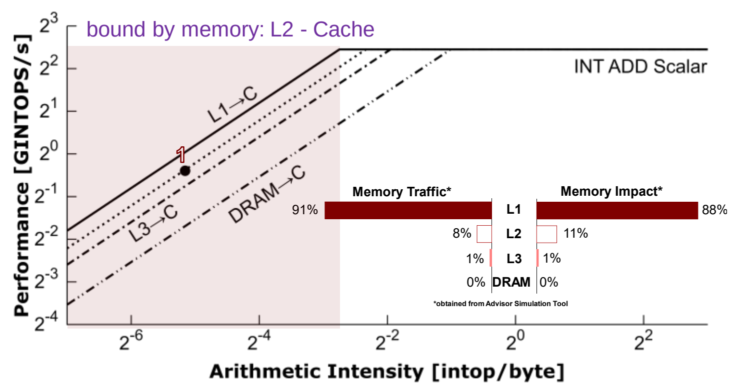


Completely dominated by Scalar instructions

Bayesian K2 Score

Original Implementation
Intel Advisor CARM





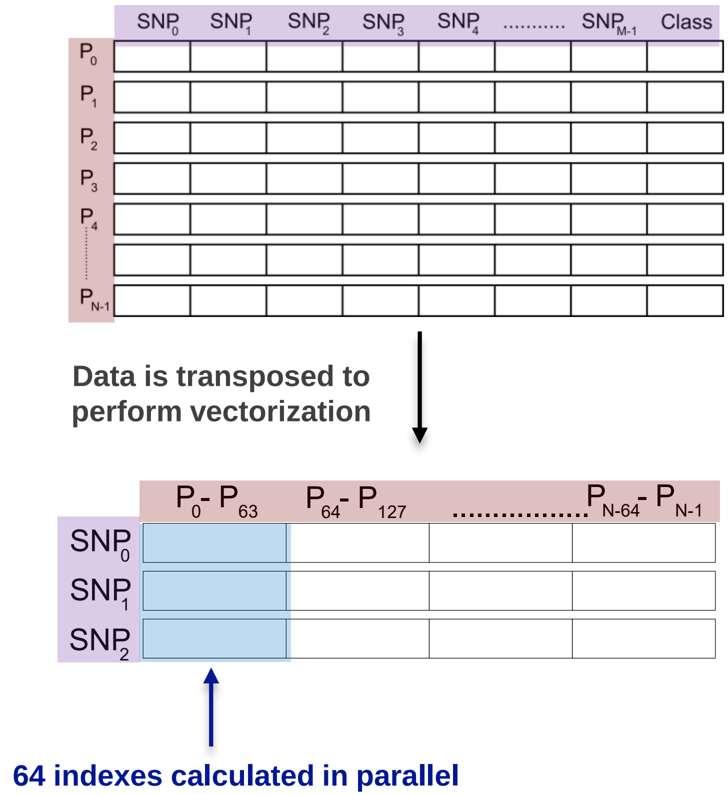
Bayesian K2 Score
High-Order Epistasis

- 1 main hotspot

Intel Advisor CARM:
- **Loop is bound by both (mem and comp)**
- Mainly limited by L3

Derived Optimizations:
- Utilization of 8-bit integers to reduce memory footprint
- Vectorization

Bayesian K2 Score

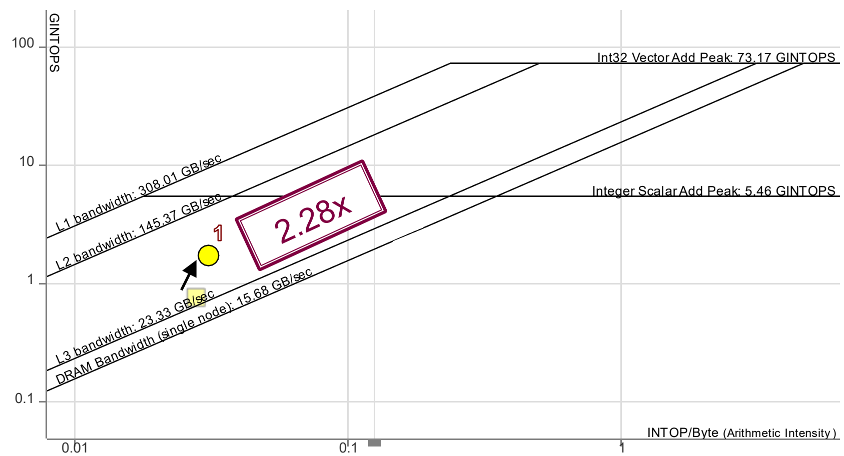


Derived Optimizations:

- Utilization of 8-bit integers to reduce memory footprint
- Vectorization

Bayesian K2 Score

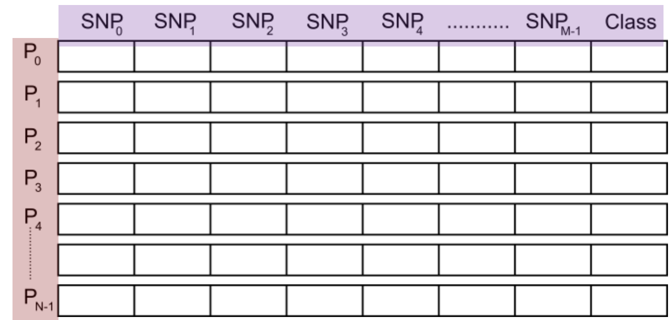
Optimized Version – Single-Thread
Intel Advisor CARM



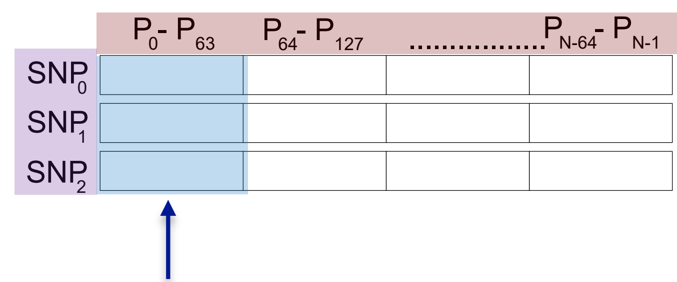
Bayesian K2 Score
High-Order Epistasis

- 1 main hotspot

Intel Advisor CARM:
- **Loop is Bound by memory**
- Between L2 and L3



Data is transposed to perform vectorization

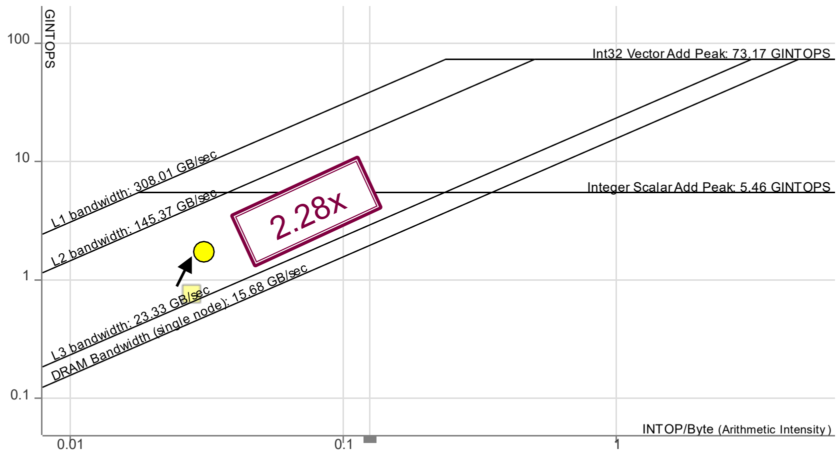


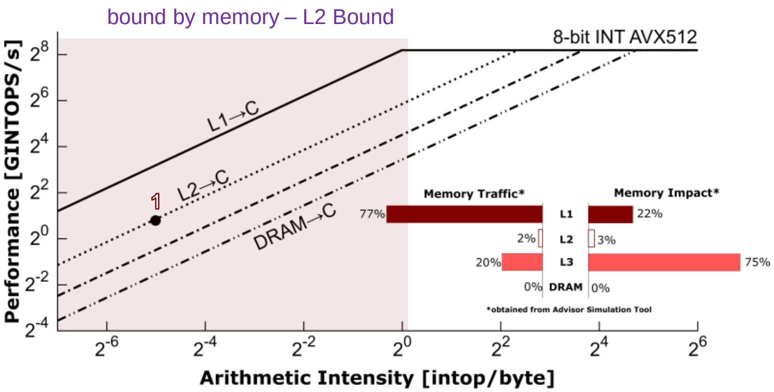
64 indexes calculated in parallel

Derived Optimizations:
- Utilization of 8-bit integers to reduce memory footprint
- Vectorization

Bayesian K2 Score

Optimized Version – Single-Thread
Intel Advisor CARM





Bayesian K2 Score

High-Order Epistasis

- 1 main hotspot

Intel Advisor CARM:

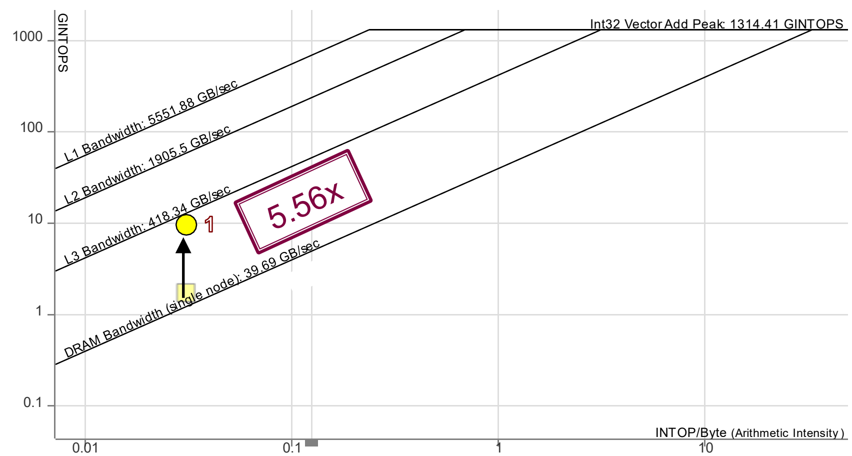
- **Loop is Bound by memory**
- Between L2 and L3

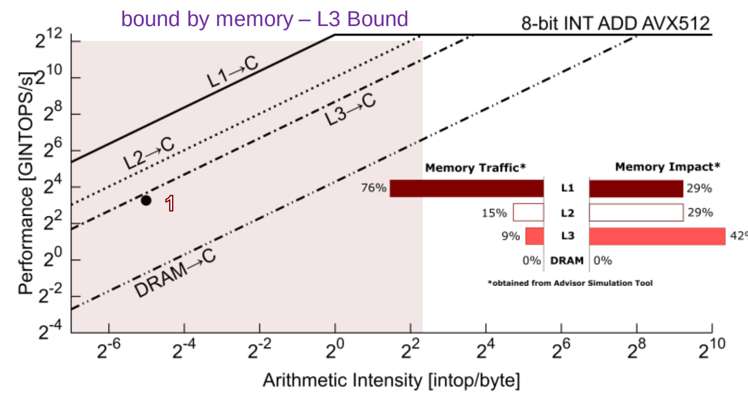
Derived Optimizations:

- Utilization of 8-bit integers to reduce memory footprint
- Vectorization

Bayesian K2 Score

Optimized Version – 18 Threads
Intel Advisor CARM





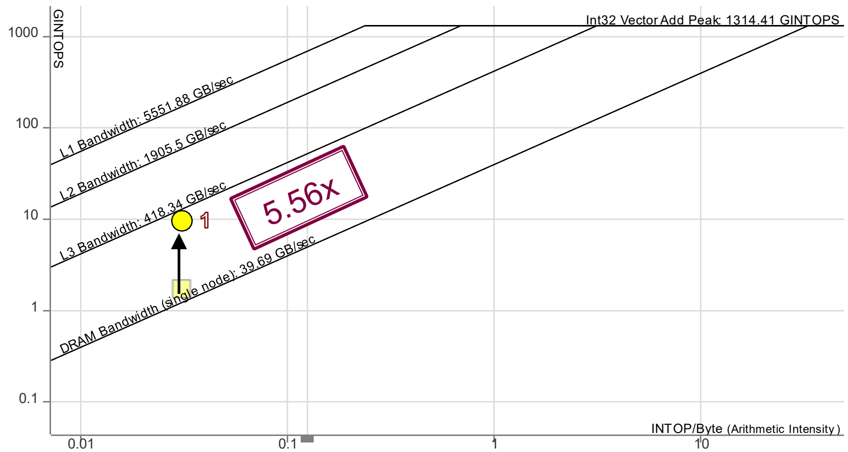
Bayesian K2 Score
High-Order Epistasis

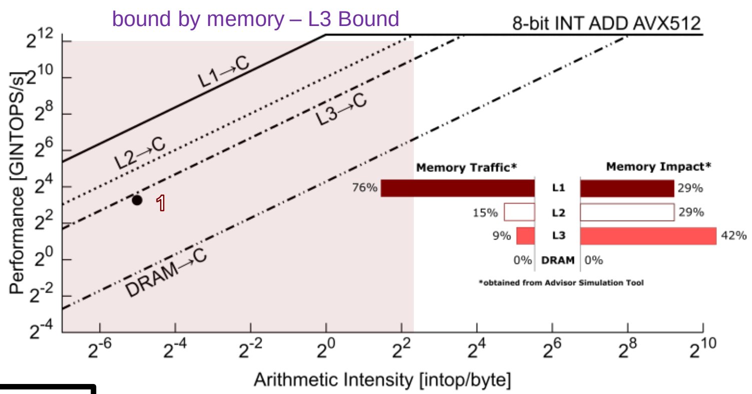
- 1 main hotspot

Intel Advisor CARM:
- **Loop is Bound by memory**
- Between L3 and DRAM (closer to L3)

Bayesian K2 Score

Optimized Version – 18 Threads
Intel Advisor CARM





Bayesian K2 Score

High-Order Epistasis

- 1 main hotspot

Intel Advisor CARM:

- **Loop is Bound by memory**
- Between L3 and DRAM (closer to L3)

Speedups:

Input Set	Order	Optimized – 1T	Optimized – 18T
1000s, 4000p	2	**23.71**	**46.54**
	3	**56.6**	**959.08**

Outline

RECAP: ORIGINAL ROOFLINE MODEL*

CACHE-AWARE ROOFLINE MODEL

- PERFORMANCE
- EXTENSIONS: POWER AND ENERGY-EFFICIENCY

APPLICATION CHARACTERIZATION WITH CACHE-AWARE ROOFLINE MODEL

- APPLICATION-DRIVEN CARM
- ISO3DFD CASE STUDY
- PROXY-APPLICATIONS FROM EXASCALE COMPUTING PROJECT
- BAYESIAN K2 SCORE

ON-GOING (FUNDED) PROJECTS

EUROPEAN
PROCESSOR
INITIATIVE

FRAMEWORK PARTNERSHIP AGREEMENT IN EUROPEAN LOW-POWER MICROPROCESSOR TECHNOLOGIES

THIS PROJECT HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S HORIZON 2020 RESEARCH AND INNOVATION PROGRAMME UNDER GRANT AGREEMENT NO 826647

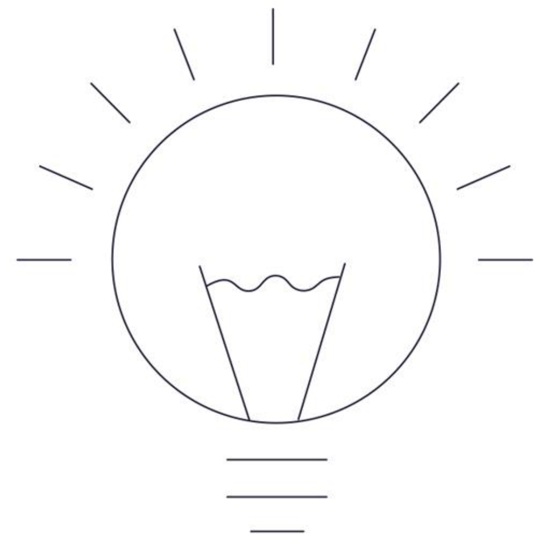
How EuroHPC will help to make us stronger

- Developing a new European supercomputing ecosystem: HPC systems, network, software, applications, access through the cloud
- Making HPC resources available to public and private users, including SMEs.
- Stimulating a technology supply industry

EuroHPC
Joint Undertaking

EPI: VISION

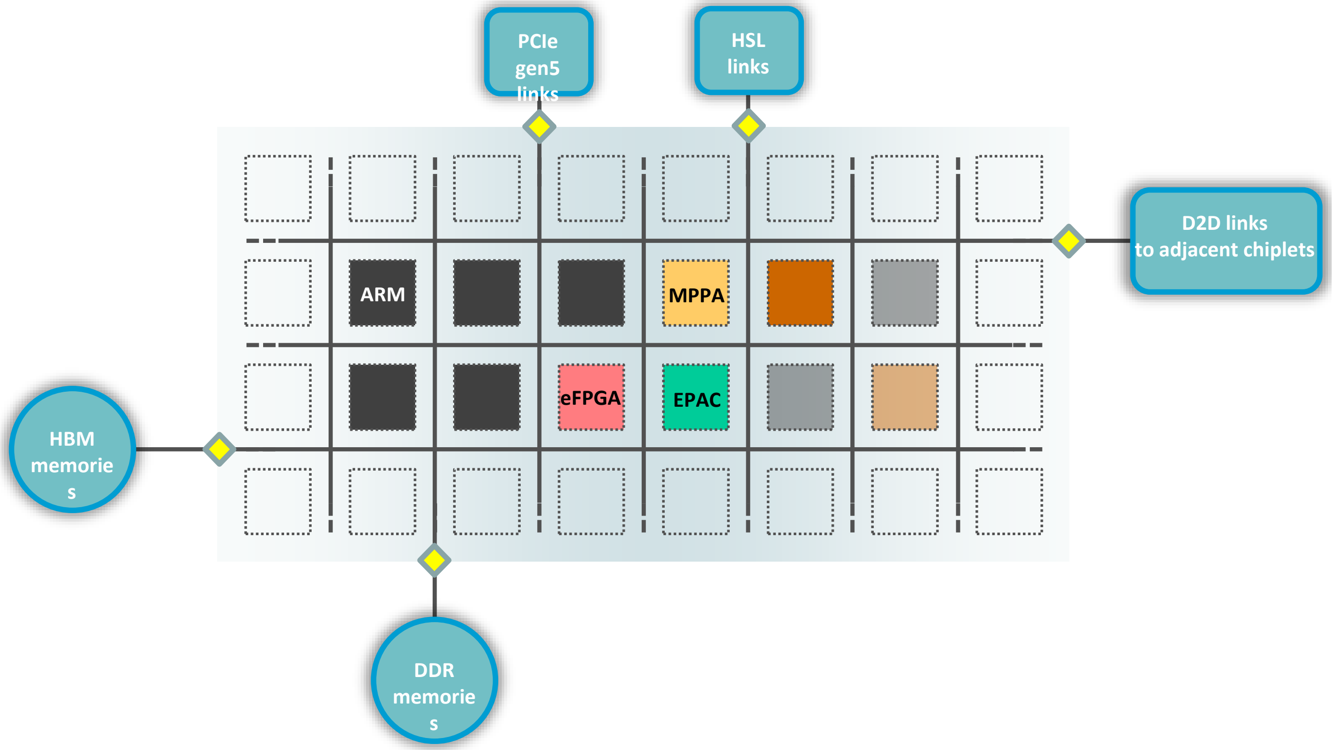
- High Performance Computing needs for Exascale machines beyond 2022
- Connected mobility & Autonomous Driving computing needs beyond 2023
- Low power CPU needs for Servers and Cloud
- Other markets under exploration (Server and Cloud)



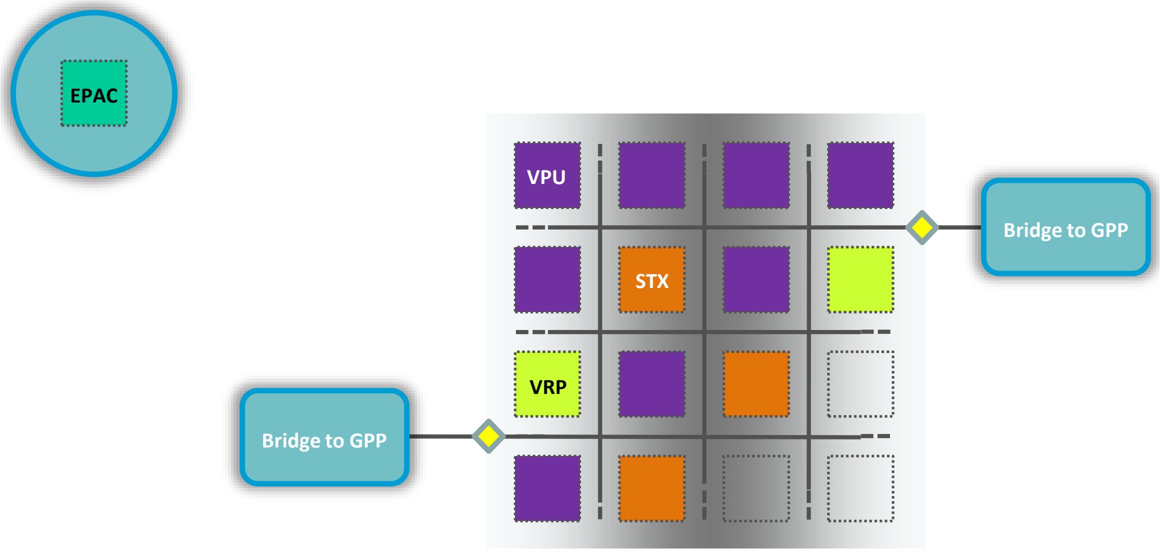
European Processor Initiative

- High Performance General Purpose Processor for HPC
- High-performance RISC-V based accelerator
- Computing platform for autonomous cars
- Will also target the AI, Big Data and other markets in order to be economically sustainable

GPP and common architecture



EPAC – RISC-V Accelerator



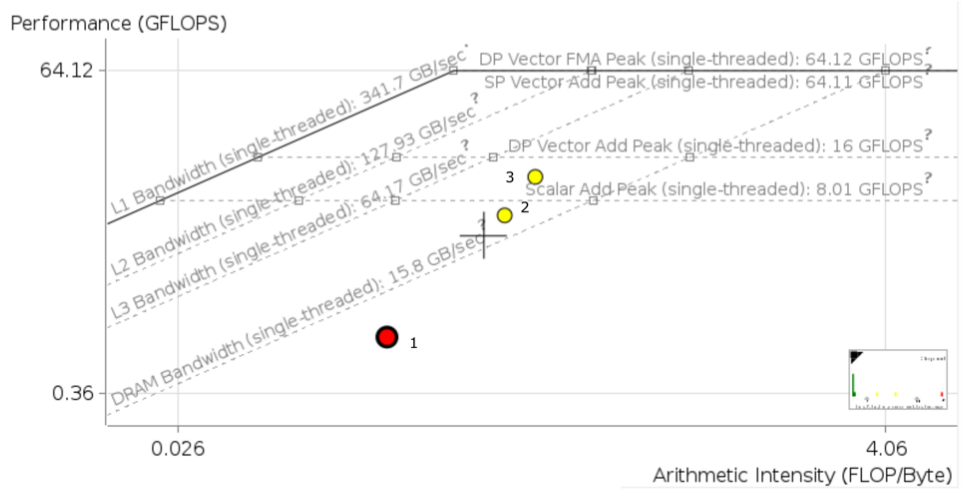
- EPAC - EPI Accelerator
- VPU – Vector Processing Unit
- STX – Stencil/Tensor accelerator
- VRP - VaRiable Precision co-processor

EPI PARTNERS

Intel Research Grant

- **Boosting the roofline-based optimization guidance and performance modeling for modern CPU systems**

- Started in 2018 and extended for 2019



Conclusions

Several Cache-aware Roofline Models (experimentally verified)

– **For several domains:** performance, power and energy

Application-Driven CARM

– **Not only considers different micro-architectures (**GPU and NUMA) but also **considers application requirements**: ISA extensions, load/store ratio
– **New visual aids for improved characterization**: memory traffic and performance impact metrics
– **Improved characterization of 3 applications representative of real-world scenarios**: ISO-3DFD, SW4Lite and Bayesean K2 score

On-going (Funded) projects

– **EPI and Intel: show the practical interest of the work**

Future work

- Include additional information in application driven CARM (integers, conversions…)

Questions?

Thank you!

Further readings:

A. Ilic, F. Pratas, and L. Sousa, ***"Beyond the Roofline: Cache-aware Power and Energy-Efficiency Modeling for Multi-cores"****, IEEE Transactions on Computers (2017)*

A. Ilic, F. Pratas, and L. Sousa, ***"Cache-aware Roofline model: Upgrading the loft",*** IEEE Computer Architecture Letters, CAL (2014)

D. Marques, H. Duarte, A. Ilic, L. Sousa, R. Belenov, P. Thierry and Z. Matveev, ***"Performance Analysis with Cache-Aware Roofline Model in Intel Advisor",*** *tutorial paper,* HPCS (2017)

A. Lopes, F. Pratas, L. Sousa and A. Ilic, ***"Exploring GPU performance, power and energy-efficiency bounds with CARM",*** ISPASS (2017)

N. Denoyelle, B. Goglin, A. Ilic, E. Jeannot and L. Sousa, ***"Modeling Non-Uniform Memory Access on Large Compute Nodes with the Cache-Aware Roofline Model",*** IEEE TPDS (2018)

A. Ilic, F. Pratas, and L. Sousa, ***"CARM: Cache-Aware Performance, Power and Energy-Efficiency Roofline Modeling",*** Intel CATC (2015)

L. Taniça, A. Ilic, P. Tomás, and L. Sousa, ***"SchedMon: A Performance and Energy Monitoring Tool for Modern Multi-cores",*** MuCoCoS/Euro-Par (2014)

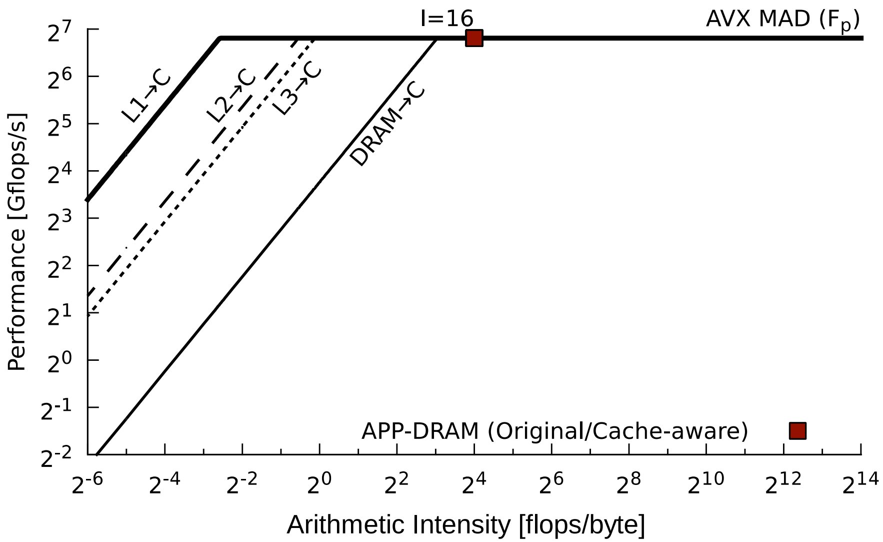
inesc id lisboa

Cache-Aware Roofline Model

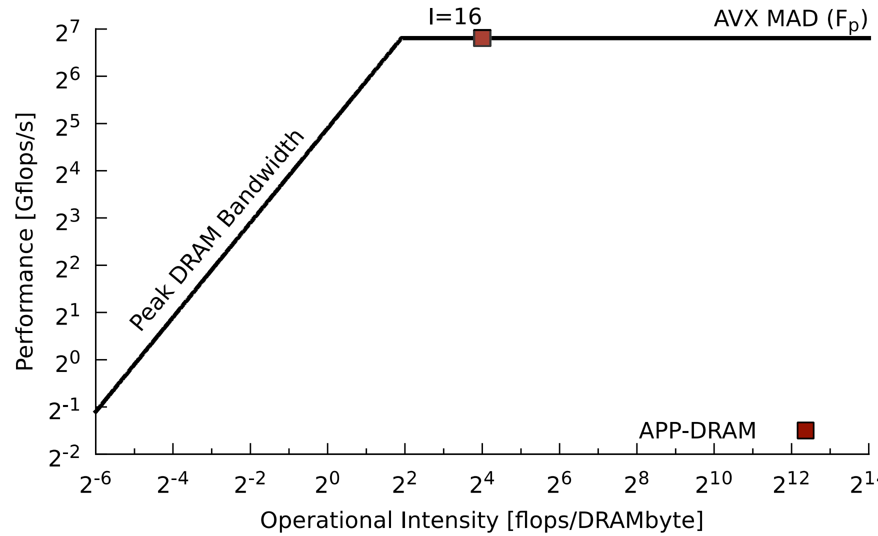
Cache-aware Roofline
Application
Characterization

Application Characterization: DRAM

Cache-aware Roofline Model (CARM)[1]



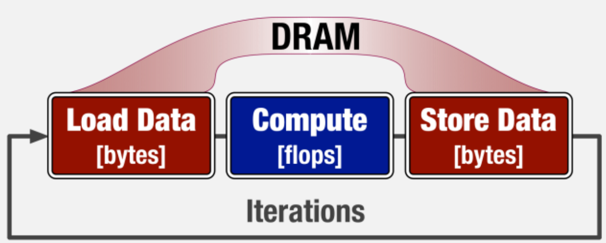
Classic DRAM Roofline Model[2]



Cache-aware Roofline Model:

iterations:	1	2	3	…	N
flops:	f	f	f	f	f
bytes:	b	b	b	b	b
operational intensity:	$\frac{f}{b}$	$\frac{2f}{2b}$	$\frac{3f}{3b}$	…	$\frac{Nf}{Nb}$

DRAM Application



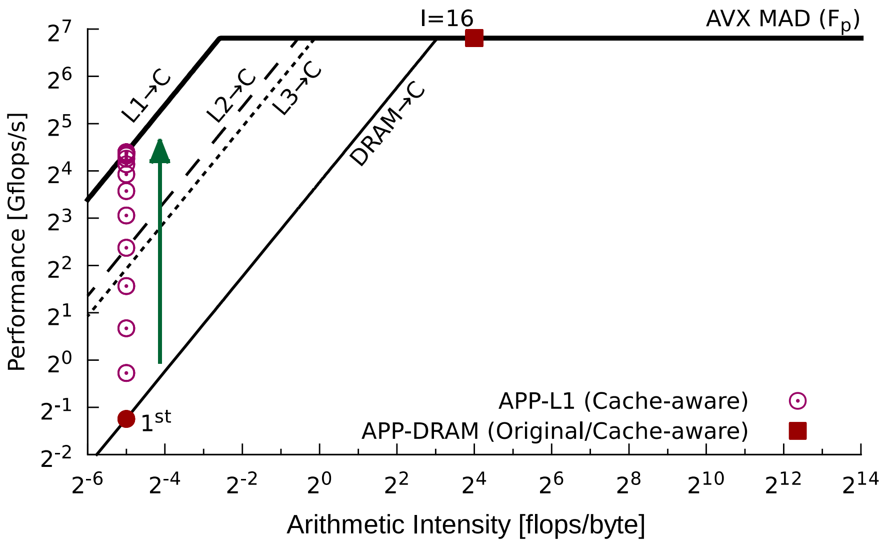
Original Roofline Model:

iterations:	1	2	3	…	N
flops:	f	f	f	f	f
DRAMbytes:	b'	b'	b'	b'	b'
operational intensity:	$\frac{f}{b'}$	$\frac{2f}{2b'}$	$\frac{3f}{3b'}$	…	$\frac{Nf}{Nb'}$

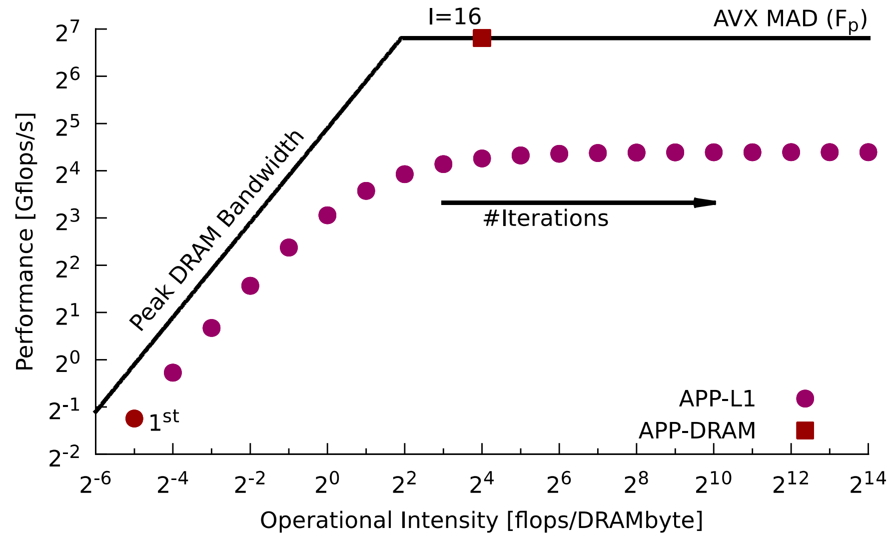
1) *Ilic A., Pratas F., Sousa L., "Cache-aware Roofline Model: Upgrading the loft", IEEE CAL (2014)*
2) *Williams S., et. al., "Roofline: An insightful visual performance model for multicore architectures", Communications of the ACM (2009)*

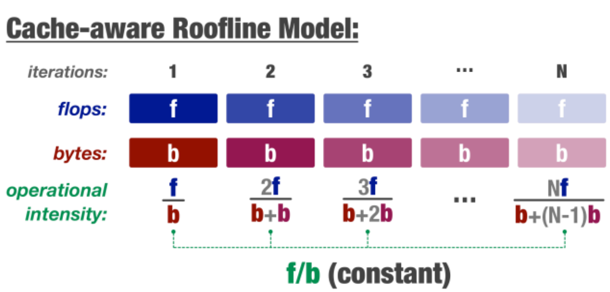
Application Characterization: L1 cache

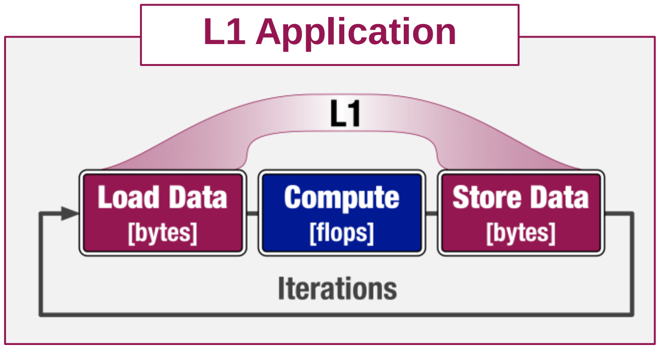
Cache-aware Roofline Model (CARM)[1]

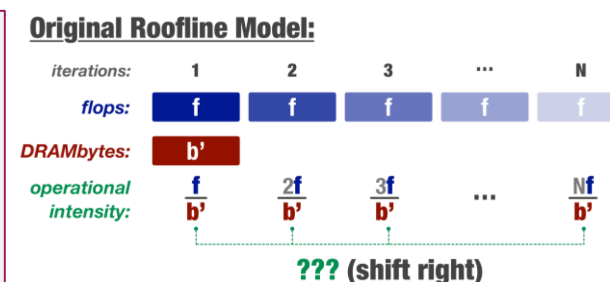


Classic DRAM Roofline Model[2]





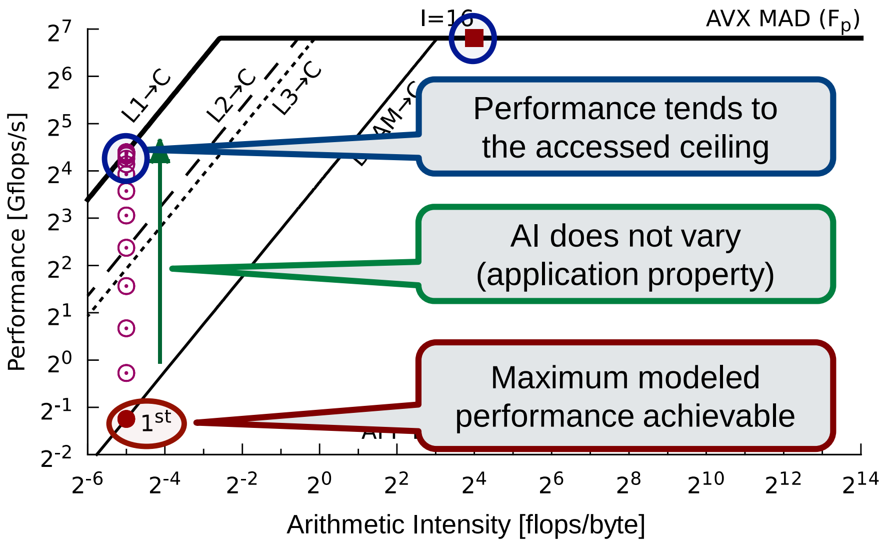




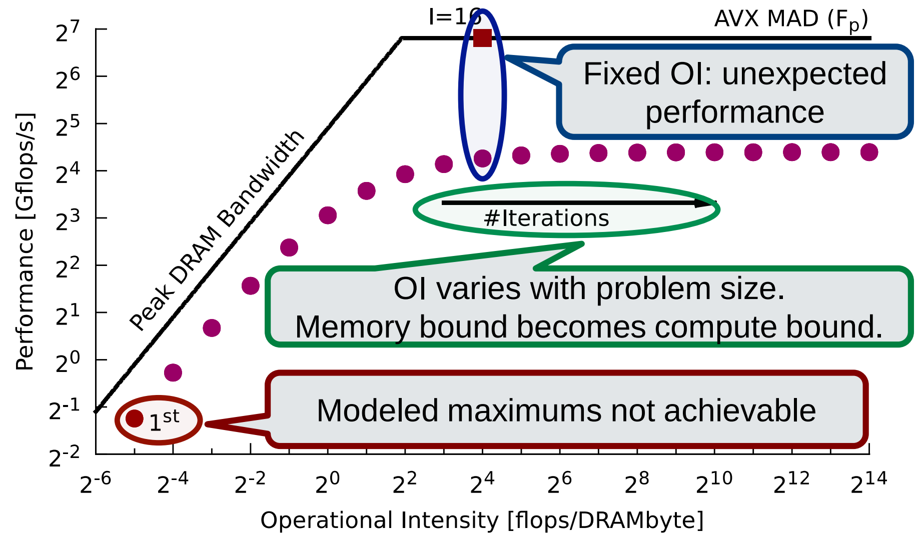
1) Ilic A., Pratas F., Sousa L., "Cache-aware Roofline Model: Upgrading the loft", IEEE CAL (2014)
2) Williams S., et. al., "Roofline: An insightful visual performance model for multicore architectures", Communications of the ACM (2009)

Application Characterization: CARM and ORM

Cache-aware Roofline Model (CARM)[1]



Classic DRAM Roofline Model[2]



Observations (Paradoxes)	Cache-aware Roofline Model (CARM)	Original Roofline Model (ORM)
Memory-bound Region: Modeled max. performance	**Achievable** (in practice)	**Not achievable** (for architectures with caches)
Intensity	**Constant** (no variation with the problem size)	**Varies with the problem size** (shift from memory-bound to compute-bound)
Characterization: Optimization Hints	**Consistent** (according to the test nature)	**Multi-model** *Hint*: Optimize memory-bound APP-L1? APP-DRAM performs better than APP-L1