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)

---

**THIS IS ALL ABOUT BUILDING THE ROOFS**
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

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 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**

- **$F_p$**: Performance (Flops/s)
- **$B_{LLC}$**: LLC Bandwidth
- **$B_{DRAM}$**: DRAM Bandwidth

### What to measure?

- **Computation (flops)**
- **Communication (L3/bytes)**
- **Communication (DRAM/bytes)**

---

**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

---

9/10/2019
**Original Roofline Model**

**Architecture Parameters**

- **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

---

9/10/2019
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: 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)
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

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 $B_{L1->C}$ directly derivable from data sheets)

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\(^3\)

\[\begin{align*}
\text{// Configured Counters} & \\
\text{CPU_CLK_UNHALTED.CORE/REF} & \\
\text{MEM_UOP_RETIRED.ALL_LOADS} & \\
\text{MEM_UOP_RETIRED.ALL_STORES} & \\
\ldots & \\
\end{align*}\]

\[\begin{align*}
\text{// AVX Assembly: 2LD+1ST} & \\
vmovapd & 0(\%rax), \%ymm0 \\
vmovapd & 32(\%rax), \%ymm1 \\
vmovapd & %ymm2, 64(\%rax) \\
vmovapd & 96(\%rax), %ymm3 \\
vmovapd & 128(\%rax), %ymm4 \\
vmovapd & %ymm5, 160(\%rax) \\
\ldots & \\
\end{align*}\]

Cache-aware Roofline Model

- 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

- 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

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)

Cache-aware Roofline Model in Intel Advisor

- **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*

---

Efficiency CARMs

**Power-efficiency**

- 4 Cores
- AVX MAD

**Energy**

- Intel 3770K Ivy Bridge
- AVX MAD (4 cores)

**Energy-efficiency**

- Maximum Efficiency AVX MAD (4 cores)
- AVX ADD/MUL

**EDP-efficiency**

- AVX MAD (4 cores)
Cache-aware Roofline Model: Use Cases

Application Characterization

Online Monitoring

* 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

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

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 CARM

State-of-the-art CARM

bound by both memory and compute? (quite hard to optimize)
Application-driven CARM (SKL-X)

State-of-the-art CARM

bound by both memory and compute?

(quiet hard to optimize)

FMA ADD DIV FMA ADD DIV FMA ADD DIV FMA ADD DIV

DP AVX-512 AVX2 SSE Scalar

SP

L1 L2 L3 DRAM L1 L2 L3 DRAM L1 L2 L3 DRAM L1 L2 L3 DRAM

LD LD/ST 2LD/ST

Memory Bandwidth [GB/s]

L1 L2 L3 DRAM L1 L2 L3 DRAM L1 L2 L3 DRAM L1 L2 L3 DRAM

2LD/ST

Scalar SP Scalar DP

FP Performance [GFLOPS/s]
ISO-3DFD Case study

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


**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)

bound by both memory and compute? (quite hard to optimize)
Preliminary Outcomes:
ISO-3DFD Case study

Intel Advisor CARM
(product release version)

Memory: L3 (DRAM?)
Compute: Scalar roof

What is my bottleneck?!

bound by both memory and compute?
(quite hard to optimize)
Intel Advisor CARM (product release version)

Memory: L3 (DRAM?)
Compute: Scalar roof

Integrated Roofline (Advisor experimental feature)

What is the bottleneck?! (quite hard to optimize)

multiple points for a single app
(one for each memory level, AIs displaced wrt the traffic between the memory levels)
Preliminary Outcomes: ISO-3DFD Case study

Intel Advisor CARM
(product release version)

What is my bottleneck?!

Memory: L3 (DRAM?)
Compute: Scalar roof

bound by both memory and compute?
(quite hard to optimize)

Integrated Roofline
(Advisor experimental feature)

Bottlenecks: Intersect respective rooflines
Point of minimum performance as bottleneck
(This app: Strictly compute bound)
Preliminary Outcomes: ISO-3DFD Case study

**Research Objectives and Goals**

Intel Advisor CARM (product release version)

- **Memory:** L3 (DRAM?)
- **Compute:** Scalar roof

**Integrated Roofline** (Advisor experimental feature)

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

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

- **Similar to the Integrated Roofline**
- **Bandwidth observed between memory levels**
- **(This app: Strictly compute bound)**
Preliminary Outcomes: ISO-3DFD Case study

Intel Advisor CARM (product release version)

Memory: L3 (DRAM?)
Compute: Scalar roof

bound by both memory and compute?
(quite hard to optimize)

Absolute architecture maximums
(can my application exploit those?)

Integrated Roofline (Advisor experimental feature)

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

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

Similar to the Integrated Roofline
Bandwidth observed between memory levels
(This app: Strictly compute bound)

Can we improve insightfulness by leveraging some app-specific data?!
Preliminary Outcomes: ISO-3DFD Case study

**Intel Advisor CARM** (product release version)

- Memory: L3 (DRAM?)
- Compute: Scalar roof

Bound by both memory and compute? (quite hard to optimize)

Absolute architecture maximums (can my application exploit those?)

**Integrated Roofline** (Advisor experimental feature)

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

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

---

Intel Advisor

<table>
<thead>
<tr>
<th>Memory</th>
<th>Compute</th>
<th>Mixed</th>
<th>Others</th>
</tr>
</thead>
<tbody>
<tr>
<td>34%</td>
<td>18%</td>
<td>41%</td>
<td>7%</td>
</tr>
</tbody>
</table>

Intel SDE

<table>
<thead>
<tr>
<th>Memory</th>
<th>Compute</th>
<th>Mixed</th>
<th>Others</th>
</tr>
</thead>
<tbody>
<tr>
<td>34%</td>
<td>18%</td>
<td>41%</td>
<td>7%</td>
</tr>
</tbody>
</table>

Intel VTune

<table>
<thead>
<tr>
<th>Memory</th>
<th>Compute</th>
<th>Mixed</th>
<th>Others</th>
</tr>
</thead>
<tbody>
<tr>
<td>34%</td>
<td>18%</td>
<td>41%</td>
<td>7%</td>
</tr>
</tbody>
</table>

Micro-benchmarking (counters, assembly)

<table>
<thead>
<tr>
<th>FP Scalar</th>
<th>FP Vector</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>59.5%</td>
<td>0%</td>
<td>40.5%</td>
</tr>
</tbody>
</table>

Source-code analysis

```c
for(TYPE_INTEGER i=0; i<end; i++)
{
    value = ptr[i] + x0;
    c1 = (FINITE_ADD(x, y));
    c2 = (FINITE_ADD(x, z));
    c3 = (FINITE_ADD(x, y));
    c4 = (FINITE_ADD(x, z));
    c5 = (FINITE_ADD(x, y));
    c6 = (FINITE_ADD(x, z));
}
```

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
Preliminary Outcomes: ISO-3DFD Case study

Intel Advisor CARM (product release version)

Integrated Roofline (Advisor experimental feature)

**Application-driven Rooflines**
(precise architecture modeling)

**Preserved model simplicity**
(intuitiveness, remove clutter)
Preliminary Outcomes: ISO-3DFD Case study

Intel Advisor CARM (product release version)

- Memory: L3 (DRAM?)
- Compute: Scalar roof

bound by both memory and compute? (quite hard to optimize)

Integrated Roofline (Advisor experimental feature)

- L1
- L2
- L3 DRAM

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

PROPOSED

Application-driven Rooflines (precise architecture modeling)
Preserved model simplicity (intuitiveness, remove clutter)
Improved bottleneck detection (model resembles app demands)

memory bound (focused optimization)
Preliminary Outcomes:
ISO-3DFD Case study

Intel Advisor CARM
(product release version)

Memory: L3 (DRAM?)
Compute: Scalar roof

bound by both memory and compute?
(quite hard to optimize)

Integrated Roofline
(Advisor experimental feature)

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

Application-driven Rooflines
(precise architecture modeling)

Preserved model simplicity
(intuitiveness, remove clutter)

Improved bottleneck detection
(model resembles app demands)

PROPOSED

Quite optimized 3D stencil:
Data locality (caches) boosts the performance
DRAM prevents from reaching the maximums

memory bound
(focused optimization)
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)

Application-driven Rooflines (precise architecture modeling)

Preserved model simplicity (intuitiveness, remove clutter)

Improved bottleneck detection (model resembles app demands)

PROPOSED

Quite optimized 3D stencil:
Data locality (caches) boosts the performance
DRAM prevents from reaching the maximums

Memory traffic shares
Data transfers and Bandwidth

Memory bound
(focused optimization)

* obtained from cache simulation

Work in progress
Preliminary Outcomes: ISO-3DFD Case study

Intel Advisor CARM (product release version)

Memory: L3 (DRAM?)
Compute: Scalar roof

bound by both memory and compute? (quite hard to optimize)

Integrated Roofline (Advisor experimental feature)

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

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
(leveraging other Advisor data)

Performance impact
(pinpointing the bottlenecks)

New set of visual aids
(to drive optimization)

Work in progress

Consistent characterization (eases code vectorization)
Preliminary Outcomes: ISO-3DFD Case study

Intel Advisor CARM (product release version)

bound by both memory and compute? (fundamentally compute with memory roofs)

memory bound (just before the ridge)

memory bound (no significant changes in characterization)

Scalar

AVX-512

PROPOSED

PROPOSED

Intel Advisor CARM (product release version)

bound by both memory and compute? (moves towards memory bound)

memory bound

* obtained from cache simulation
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)

strictly memory-bound

bound by both memory and compute
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?

- 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.
How to detect which genes influence traits or diseases?

Genetic Markers

<table>
<thead>
<tr>
<th>SNP_0</th>
<th>SNP_1</th>
<th>SNP_2</th>
<th>SNP_3</th>
<th>SNP_4</th>
<th>...........</th>
<th>SNP_K</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>P_0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P_1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P_2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P_3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P_4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</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.
How to detect which genes influence traits or diseases?

Bayesian K2 Score

Patients

Genetic Markers

Populated with values 0, 1 and 2

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.
How to detect which genes influence traits or diseases?

Genetic Markers

Patients

Populated with values 0, 1 and 2

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.
How to detect which genes influence traits or diseases?

K=3:

- 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.

Patients

<table>
<thead>
<tr>
<th></th>
<th>SNP₀</th>
<th>SNP₁</th>
<th>SNP₂</th>
<th>SNP₃</th>
<th>SNP₄</th>
<th>...........</th>
<th>SNPₙ₄</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>P₀</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P₁</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P₂</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P₃</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P₄</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P₅</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Populated with values 0, 1 and 2

1 - disease
0 – no disease
How to detect which genes influence traits or diseases?

K=3:

- 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.

Patients

<table>
<thead>
<tr>
<th>P_0</th>
<th>P_1</th>
<th>P_2</th>
<th>P_3</th>
<th>P_4</th>
<th>\ldots</th>
<th>P_{N-1}</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNP_0</td>
<td>SNP_1</td>
<td>SNP_2</td>
<td>SNP_3</td>
<td>SNP_4</td>
<td>\ldots</td>
<td>SNP_{M-1}</td>
<td></td>
</tr>
</tbody>
</table>

Populated with values 0, 1 and 2

1 - disease
0 - no disease
How to detect which genes influence traits or diseases?

**K=3:**

Bayesian K2 Score

- 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

**High-Order Epistasis**

 Patients

<table>
<thead>
<tr>
<th></th>
<th>SNP₁</th>
<th>SNP₂</th>
<th>SNP₃</th>
<th>SNP₄</th>
<th>...........</th>
<th>SNPₙ₄</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>P₀</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P₁</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P₂</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P₃</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P₄</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pₙ₋₁</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Populated with values 0, 1 and 2

1 - disease
0 – no disease
How to detect which genes influence traits or diseases?

**K=3:**

K2 Bayesian Score

<table>
<thead>
<tr>
<th>SNP_0</th>
<th>SNP_1</th>
<th>SNP_2</th>
<th>SNP_3</th>
<th>SNP_4</th>
<th>…………</th>
<th>SNP_M_1</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P_0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P_1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P_2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P_3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P_4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>…………</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Patients populated with values 0, 1 and 2

1 - disease
0 - no disease

**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.
How to detect which genes influence traits or diseases?

K=3:

- Bayesian K2 Score
- 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.

<table>
<thead>
<tr>
<th>Patients</th>
<th>SNP_0</th>
<th>SNP_1</th>
<th>SNP_2</th>
<th>SNP_3</th>
<th>SNP_4</th>
<th>......</th>
<th>SNP_n</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>P_0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P_1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P_2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P_3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P_4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>......</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P_N-1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Populated with values 0, 1 and 2

1 - disease
0 – no disease
How to detect which genes influence traits or diseases?

K=3: Bayesian K2 Score

Population with values 0, 1 and 2

Population frequency table

<table>
<thead>
<tr>
<th>Class</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
<th>21</th>
<th>22</th>
<th>23</th>
<th>24</th>
<th>25</th>
<th>26</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class = 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class = 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</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.
How to detect which genes influence traits or diseases?

K=3:

Populated with values 0, 1 and 2

1 - disease
0 – no disease

Populate frequency table

#Columns = 3^k

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?

K=3:

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.

Patients

Populated with values 0, 1 and 2

1 - disease
0 – no disease

Populate frequency table

#Columns = $3^k$

Class = 0

Class = 1

K2 Calculation
Bayesian K2 Score

How to detect which genes influence traits or diseases?

K=3:

- 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

High-Order Epistasis

\[
K2 = \sum_{j=1}^{x+y+1} \log(j) + \sum_{j=1}^{x} \log(j) + \sum_{j=1}^{y} \log(j)
\]
How to detect which genes influence traits or diseases?

K=3:

Patients

Populated with values 0, 1 and 2

1 - disease
0 - no disease

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:

$$K2 = \sum_{j=1}^{X+Y+1} \log (j) + \sum_{j=1}^{X} \log (j) + \sum_{j=1}^{Y} \log (j)$$
How to detect which genes influence traits or diseases?

K=3:

Population with values 0, 1 and 2

Populate frequency table

#Columns = $3^k$

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

K2 Score:

$$K2 = \sum_{j=1}^{X+Y+1} \log(j) + \sum_{j=1}^{X} \log(j) + \sum_{j=1}^{Y} \log(j)$$
How to detect which genes influence traits or diseases?

**K=3:**

- **Bayesian K2 Score**
  - 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.

Patients

Populated with values 0, 1 and 2

Populate frequency table

#Columns = \(3^k\)

Class = 0

Class = 1

**K2 Calculation**

Bayesian K2 Score

High-Order Epistasis

Index Calculation:

\[
IDX = ((1 \times 3) + 2) \times 3
\]

K2 Score:

\[
K2 = \sum_{j=1}^{X+Y+1} \log(j) + \sum_{j=1}^{X} \log(j) + \sum_{j=1}^{Y} \log(j)
\]
How to detect which genes influence traits or diseases?

**K=3:**

Patients populated with values 0, 1, and 2. 

- Disease: 1
- No disease: 0

Populate frequency table:

#Columns = \(3^k\)

- Index Calculation:
  \[
  IDX = ((1 \times 3 + 2) \times 3 + 0)
  \]

- 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

- 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

Original Implementation
Intel Advisor CARM

Index Calculation + Frequency Table

Completely dominated by Scalar instructions

Advisor Instruction Mix

82.985s
Scalar
13.166s
Total time
Self time

- 1 main hotspot

Intel Advisor CARM:
- Loops is bound by both (mem and comp)
- Loop 1: Mainly limited by L3

High-Order Epistasis

9/10/2019
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

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

**High-Order Epistasis**

- 1 main hotspot

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

---

Optimized Version – Single-Thread
Intel Advisor CARM

![Graph showing performance improvement](image)

Data is transposed to perform vectorization

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

High-Order Epistasis

- 1 main hotspot

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

<table>
<thead>
<tr>
<th>Input Set</th>
<th>Order</th>
<th>Optimized – 1T</th>
<th>Optimized – 18T</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000s, 4000p</td>
<td>2</td>
<td>23.71</td>
<td>46.54</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>56.6</td>
<td>959.08</td>
</tr>
</tbody>
</table>
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
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
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

- ARM
- MPPA
- eFPGA
- EPAC
- PCIe gen5 links
- HSL links
- D2D links to adjacent chiplets
- HBM memories
- DDR memories
EPAC – RISC-V Accelerator

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

9/10/2019
Questions?
Thank you!

Further readings:


Cache-Aware Roofline Model

Cache-aware Roofline Application Characterization
Application Characterization:
DRAM

**Cache-aware Roofline Model (CARM)**

- Graph comparing performance against arithmetic intensity for different cache levels and DRAM access times.

**Classic DRAM Roofline Model**

- Graph showing performance against operational intensity for DRAM bandwidth.

---

**Cache-aware Roofline Model:**

- Iterations: 1, 2, 3, ..., N
- Flops: f, f, f, ..., f
- Bytes: b, b, b, ..., b
- Operational intensity: f, 2f, 3f, ..., Nf

**DRAM Application**

1. **Load Data** [bytes]
2. **Compute** [flops]
3. **Store Data** [bytes]

---

**Original Roofline Model:**

- Iterations: 1, 2, 3, ..., N
- Flops: f, f, f, ..., f
- DRAMBytes: b', b', b', b', ..., b'
- Operational intensity: f, 2f, 3f, ..., Nf

---

Application Characterization: L1 cache

Cache-aware Roofline Model (CARM)\(^1\)

Classic DRAM Roofline Model\(^2\)

### Cache-aware Roofline Model:

- **Arithmetic Intensity [flops/byte]**
- **Performance [GFlops/s]**
- **L1 cache**
- **L1+C**
- **L2+C**
- **L3+C**
- **DRAM+C**
- **L1 Application**
- **APP-L1 (Cache-aware)**
- **APP-DRAM (Original/Cache-aware)**
- **AVX MAD (F₀)**

### Classic DRAM Roofline Model:

- **Performance [GFlops/s]**
- **Operational Intensity [flops/DRAMbyte]**
- **Peak DRAM Bandwidth**
- **#Iterations**
- **APP-L1**
- **APP-DRAM**
- **L1 Application**

### Cache-aware Roofline Model:

- **Iterations:** 1, 2, 3, ..., N
- **Flops:** f, f, f, f, f, f
- **Bytes:** b, b, b, b, b, b
- **Operational Intensity:** f, b, b, b, b, b

### Original Roofline Model:

- **Iterations:** 1, 2, 3, ..., N
- **Flops:** f, f, f, f, f, f
- **DRAMBytes:** b', b', b', b', b', b'
- **Operational Intensity:** f, b', b', b', b', b'

---

Application Characterization: CARM and ORM

**Cache-aware Roofline Model (CARM)\(^1\)**

- **Performance tends to the accessed ceiling**
- **AI does not vary** (application property)
- **Maximum modeled performance achievable**

<table>
<thead>
<tr>
<th>Observations (Paradoxes)</th>
<th><strong>CACHE-AWARE ROOFLINE MODEL (CARM)</strong></th>
<th><strong>ORIGINAL ROOFLINE MODEL (ORM)</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MEMORY-BOUND REGION:</strong> Modeled max. performance</td>
<td>Achievable (in practice)</td>
<td>Not achievable (for architectures with caches)</td>
</tr>
<tr>
<td><strong>INTENSITY</strong></td>
<td>Constant (no variation with the problem size)</td>
<td>Varies with the problem size (shift from memory-bound to compute-bound)</td>
</tr>
<tr>
<td><strong>CHARACTERIZATION:</strong> Optimization Hints</td>
<td>Consistent (according to the test nature)</td>
<td>Multi-model</td>
</tr>
</tbody>
</table>