Performance Engineering by Means of Automated Performance Modeling

with

Kerncraft

Released under AGPLv3 on
github.com/RRZE-HPC/kerncraft

Vision:
Automatically predict single-node loop kernel performance
on current and future architectures.

Julian Hammer <julian.hammer@fau.de>, Friedrich-Alexander-University Erlangen-Nuremberg

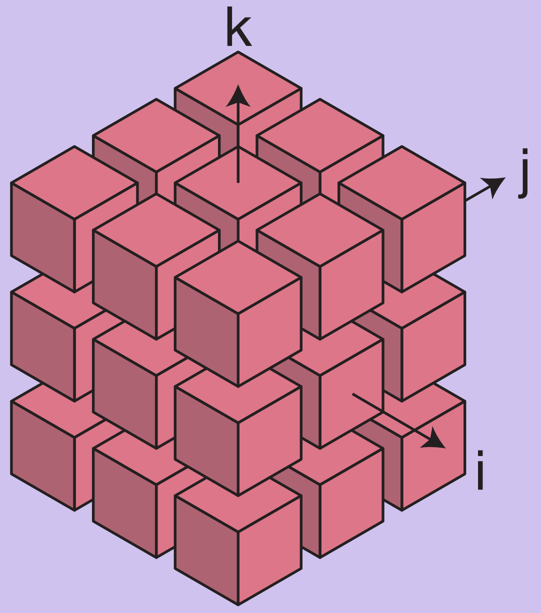
Kernel

Required computational and data resources

Restrictions:
> no branches in inner loop body
> only affine loops and array indices
typically for dens linear algebra, stencil, streaming codes

C Code (stable)

```c
double a[M][N][N], b[M][N][N], s;
for(int k=1; k<M-1; ++k)
  for(int j=1; j<N-1; ++j)
    for(int i=1; i<N-1; ++i)
      b[k][j][i] = (
        (a[k][j][i] + a[k][j][i-1] + a[k][j][i+1]) +
        (a[k][j-1][i] + a[k][j-1][i-1] + a[k-1][j][i-1]) +
        (a[k][j+1][i] + a[k][j+1][i-1] + a[k][j+1][i+1]) +
        (a[k-1][j][i] + a[k][j-1][i+1] + a[k-1][j][i+1]) +
        (a[k+1][j][i] + a[k+1][j][i-1] + a[k+1][j][i+1]) +
        (a[k-1][j-1][i] + a[k-1][j-1][i-1] + a[k-1][j-1][i+1]) +
        (a[k-1][j+1][i] + a[k-1][j+1][i-1] + a[k-1][j+1][i+1]) +
        (a[k+1][j-1][i] + a[k+1][j-1][i-1] + a[k+1][j-1][i+1]) +
        (a[k-1][j+1][i+1] + a[k+1][j+1][i] + a[k-1][j+1][i-1])) * s;
```



OR

LLVM IR/Polly (in-development)

LLVM-Polly finds suitable loop nests and exports them

OR

Metadata (prototype)

High-level description of loop nest resources

Machine

Offered computational and data resources

Semi-Automatic Gathering (prototype)

YAML based description of hardware architecture

Ivy-Bridge Machine configuration file:

CPU & compiler

```
model name: Intel(R) Xeon(R) CPU
            E5-2660 v2 @ 2.20GHz
compiler: icc: -O3 -xAVX -fno-alias
```

Memory subsystem

```
memory hierarchy:
    - level: L1
      cache per group:
         {'sets': 64, 'ways': 8,
          'cl_size': 64, # 32 kB
          'replacement_policy': 'LRU',
          'write_allocate': True,
          'write_back': True,
          'load_from': 'L2',
          'store_to': 'L2'}
      cycles per cacheline transfer: 2
```

Benchmark results

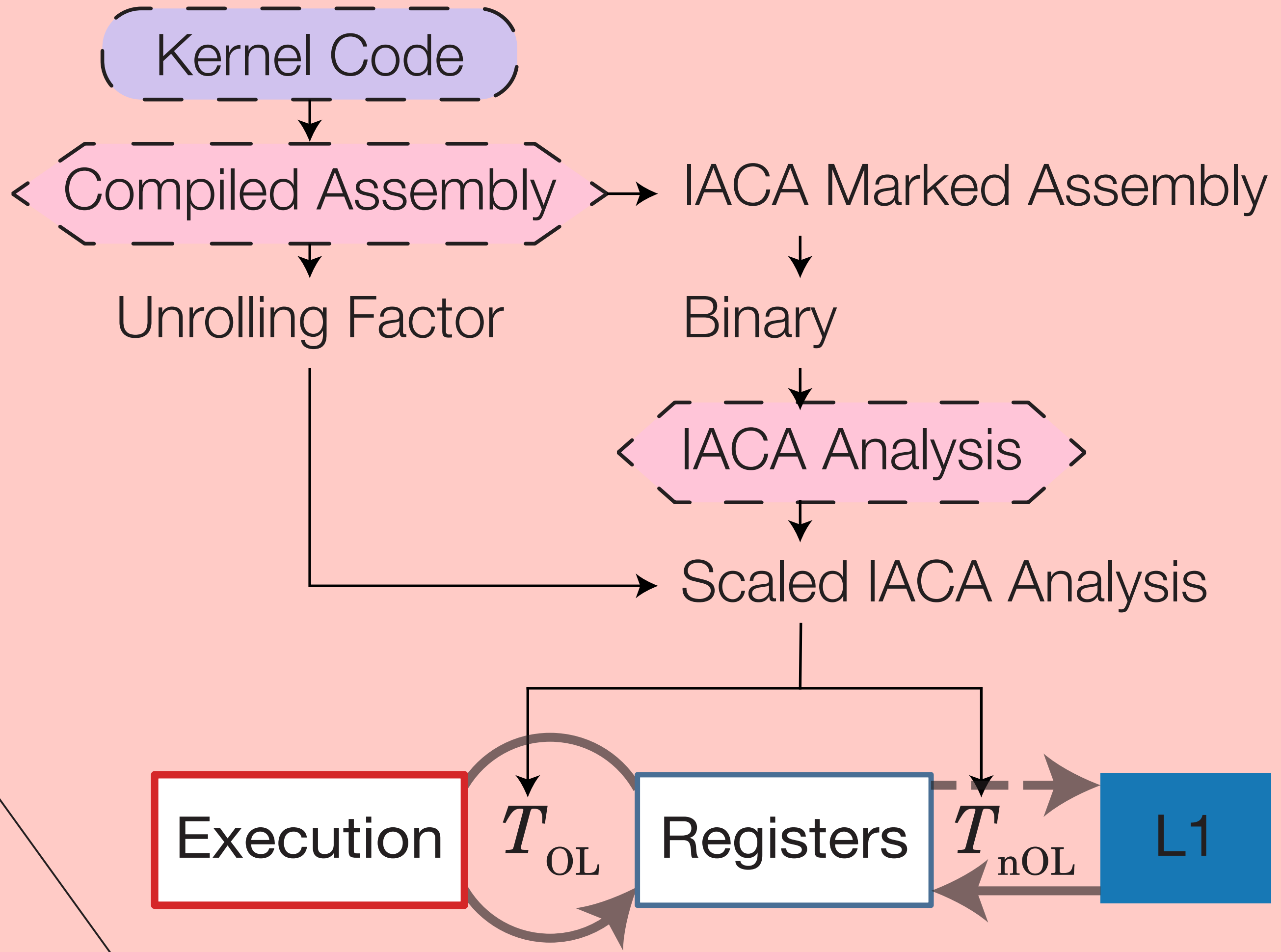
```
benchmarks: {measurements:
                {MEM: {results:
    update: [18.91 GB/s, 32.43 GB/s,
             37.28 GB/s, 39.98 GB/s, ...]}}
```

In-core prediction

Computation and loading from/storing to L1

Intel Architecture Code Analyzer (stable)

Based on Intel's IACA



OR

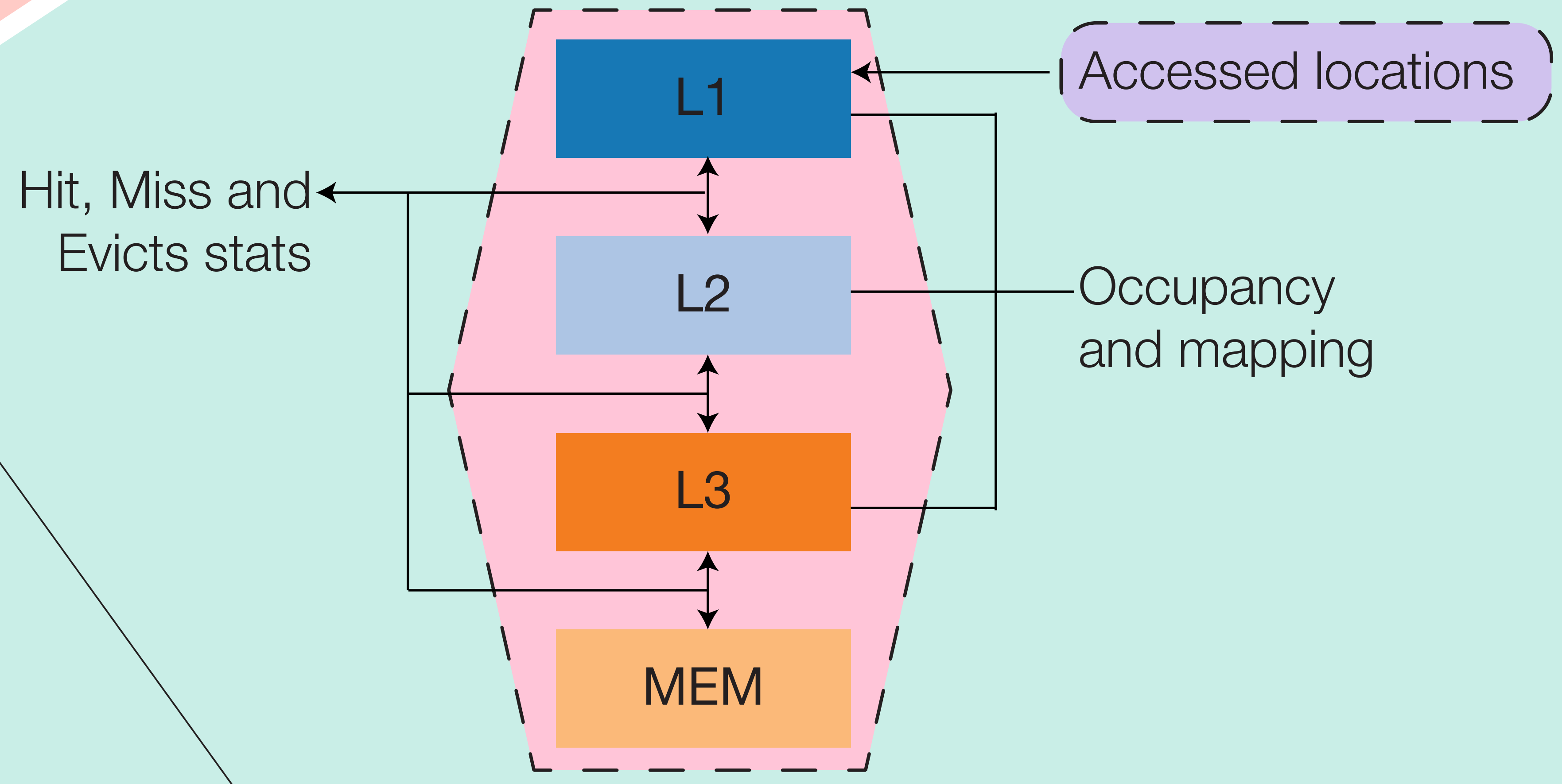
Open Source Architecture Code Analyzer (in-development)

Open source replacement for IACA

1. Gather Instructions
2. Measure latency, throughput and overlap with microbenchmarks
3. Generate in-core model
4. Apply to loop-nest assembly

Cache Simulator (stable)

Complete simulation with pycachesim

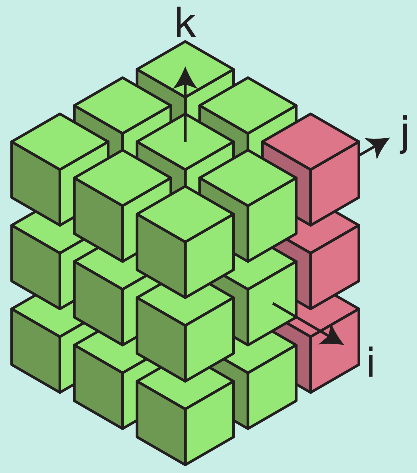


OR

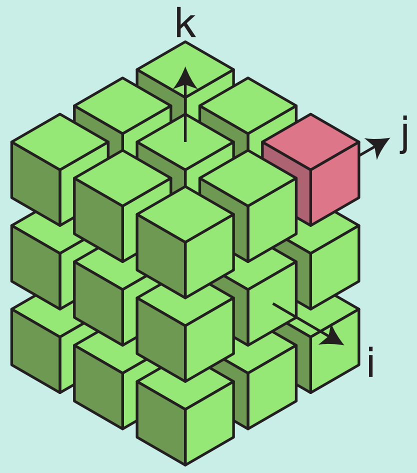
Layer Conditions (stable)

Analytical model for inclusive, LRU caches

$$C_{\mathrm{req}} = \sum(L_{\leq t}) + t * \mathrm{count}(L_{>t})$$



$t = N-2$	C_{req}	$t = N^2-2N-2$
$N \leq 410$	$\leq 32\,\mathrm{kB}$	$N \leq 32$
$N \leq 3277$	$\leq 256\,\mathrm{kB}$	$N \leq 91$
$N \leq 327680$	$\leq 25\,\mathrm{MB}$	$N \leq 905$
18	Hits	26
10	Misses	2



Cache prediction

Hits/Misses in all cache levels

Performance Model

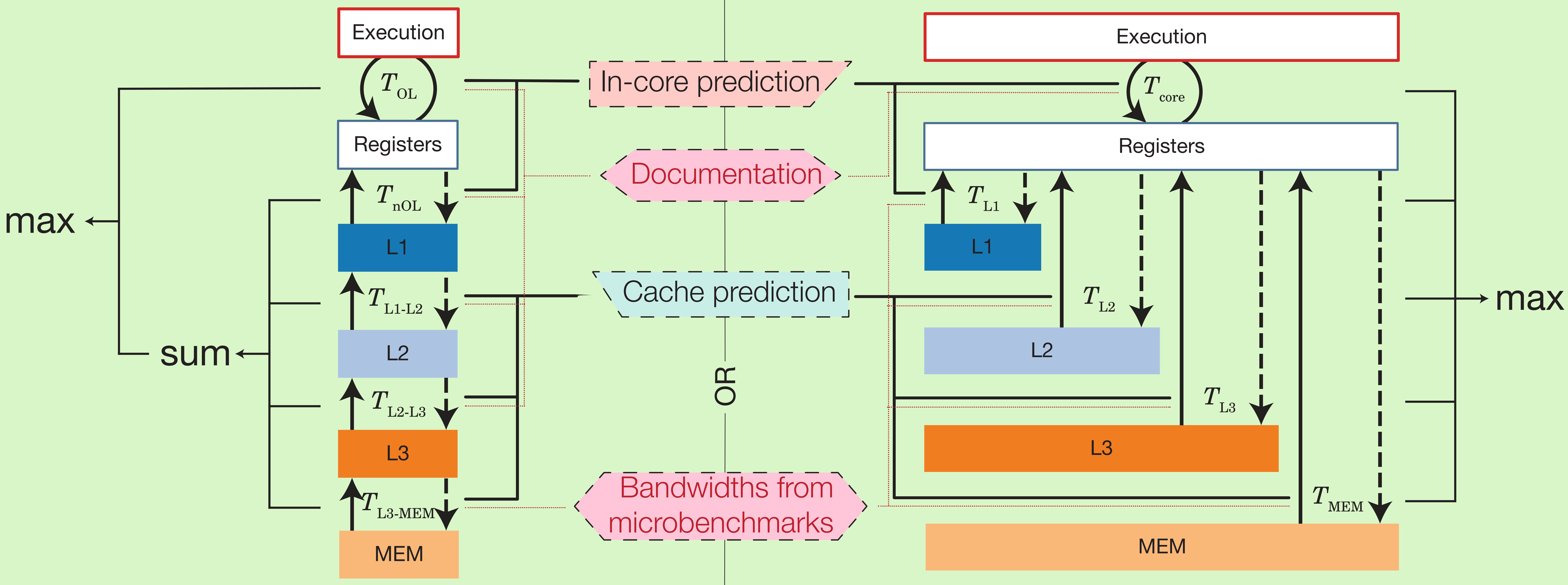
Compilation of runtime prediction

Execution-Cache-Memory (stable)

Model with serial assumption on data path and cache bandwidths based on documentation

Roofline (stable)

Model with overlap assumption on all bottlenecks, based on measured bandwidths



Conclusions

Taking (some) pain out of analytical performance modeling

Key components are relevant to:

> compilers
tile size selection

> architecture researchers
impact of changes on runtime

> offline optimizations
preevaluation of instruction mix

> energy-efficient computing
selection of cores and frequencies

> software developers
What is going on here?

Julian Hammer <julian.hammer@fau.de>

Open Source and freely available at:

github.com/RRZE-HPC/kerncraft
github.com/RRZE-HPC/pycachesim

Interactive LC Calculator:

https://rrze-hpc.github.io/layer-condition

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

Supported by

HPC

Federal Ministry
of Education
and Research

SKAMPY, METACCA &
SeaSite

DAAD
Deutscher Akademischer Austauschdienst
German Academic Exchange Service

FITweltweit