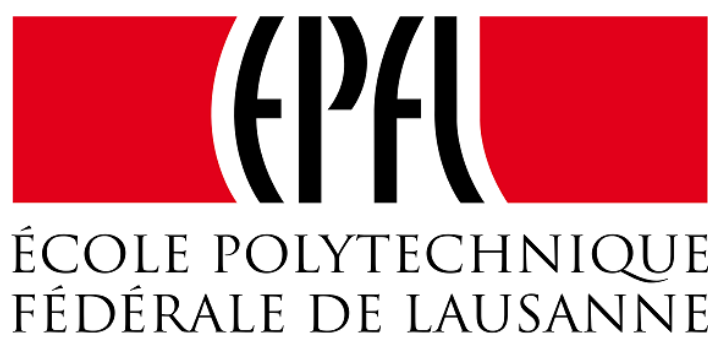
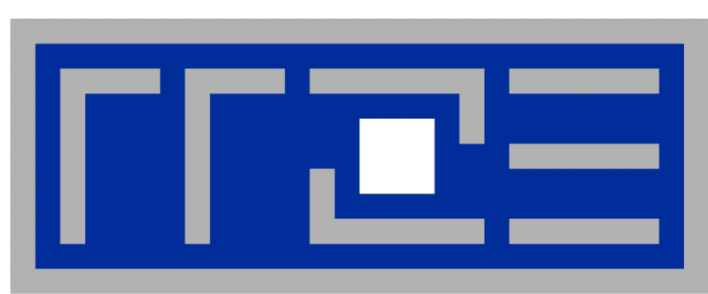




Poster artifacts



https://tiny.cc/ECM-SC18-AD

J. Hofmann et al.: On the accuracy and usefulness of analytic energy models for contemporary multicore processors. DOI: [10.1007/978-3-319-92040-5_2](https://doi.org/10.1007/978-3-319-92040-5_2)

J. Hammer et al.: Kerncraft: A Tool for Analytic Performance Modeling of Loop Kernels. DOI: [10.1007/978-3-319-56702-0_1](https://doi.org/10.1007/978-3-319-56702-0_1)

G. Hager et al.: Exploring performance and power properties of modern multicore chips via simple machine models. DOI: [10.1002/cpe.3180](https://doi.org/10.1002/cpe.3180)

H. Stengel et al.: Quantifying performance bottlenecks of stencil computations using the Execution-Cache-Memory model. DOI: [10.1145/2751205.2751240](https://doi.org/10.1145/2751205.2751240)

T. M. Malas et al.: Optimization of an electromagnetics code with multicore wavefront diamond blocking and multi-dimensional intra-tile parallelization. DOI: [10.1009/IPDPS.2016.87](https://doi.org/10.1009/IPDPS.2016.87)

T. Ewart et al.: Neuromapp: A Mini-application Framework to Improve Neural Simulators. DOI: [10.1007/978-3-319-58667-0_10](https://doi.org/10.1007/978-3-319-58667-0_10)

Layer Condition Calculator: tiny.cc/LayerConditions

Kerncraft: tiny.cc/kerncraft

Pycachesim: tiny.cc/pycachesim

IACA: tiny.cc/IACA

OSACA: tiny.cc/OSACA

LIKWID tools: tiny.cc/LIKWID

Embedded MM content



https://tiny.cc/ECM-SC18

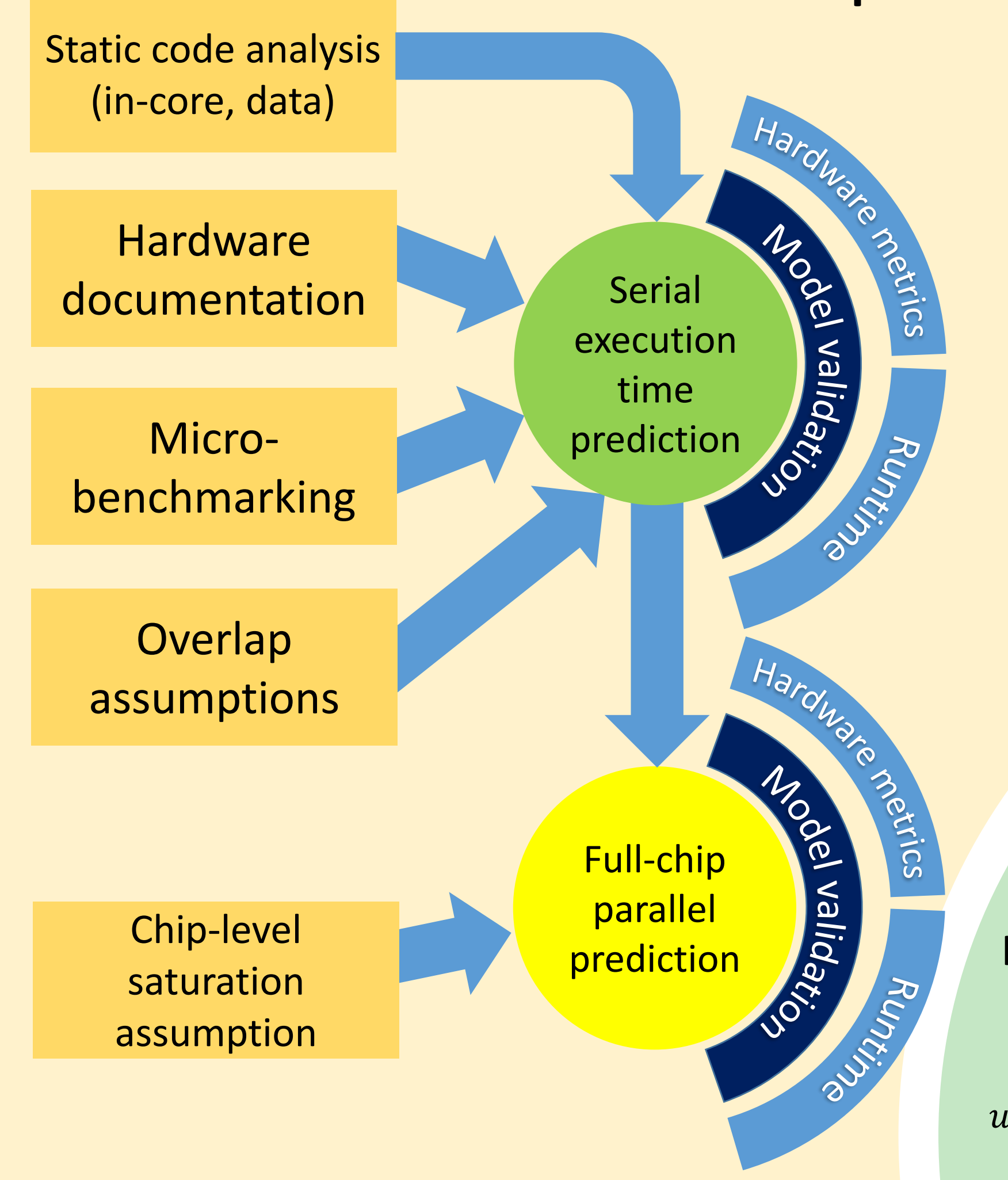
Applying the Execution-Cache-Memory Performance Model: Current State of Practice

G. Hager¹, J. Eitzinger¹, J. Hornich¹, F. Cremonesi², C. A. Alappat¹, T. Röhl¹, G. Wellein¹

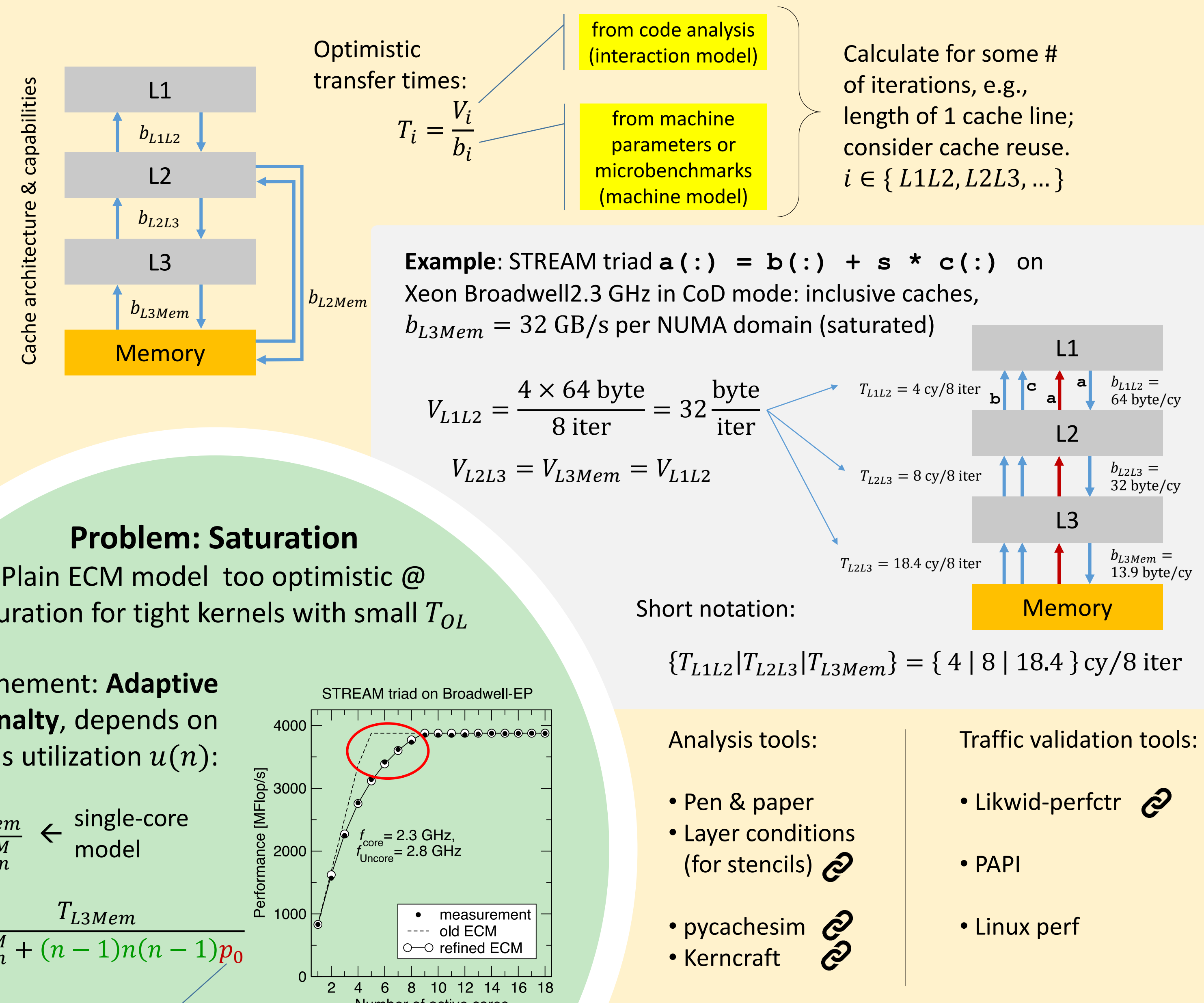
¹ Erlangen Regional Computing Center (RRZE), Erlangen, Germany

² École Polytechnique Fédérale de Lausanne (EPFL), Geneva, Switzerland

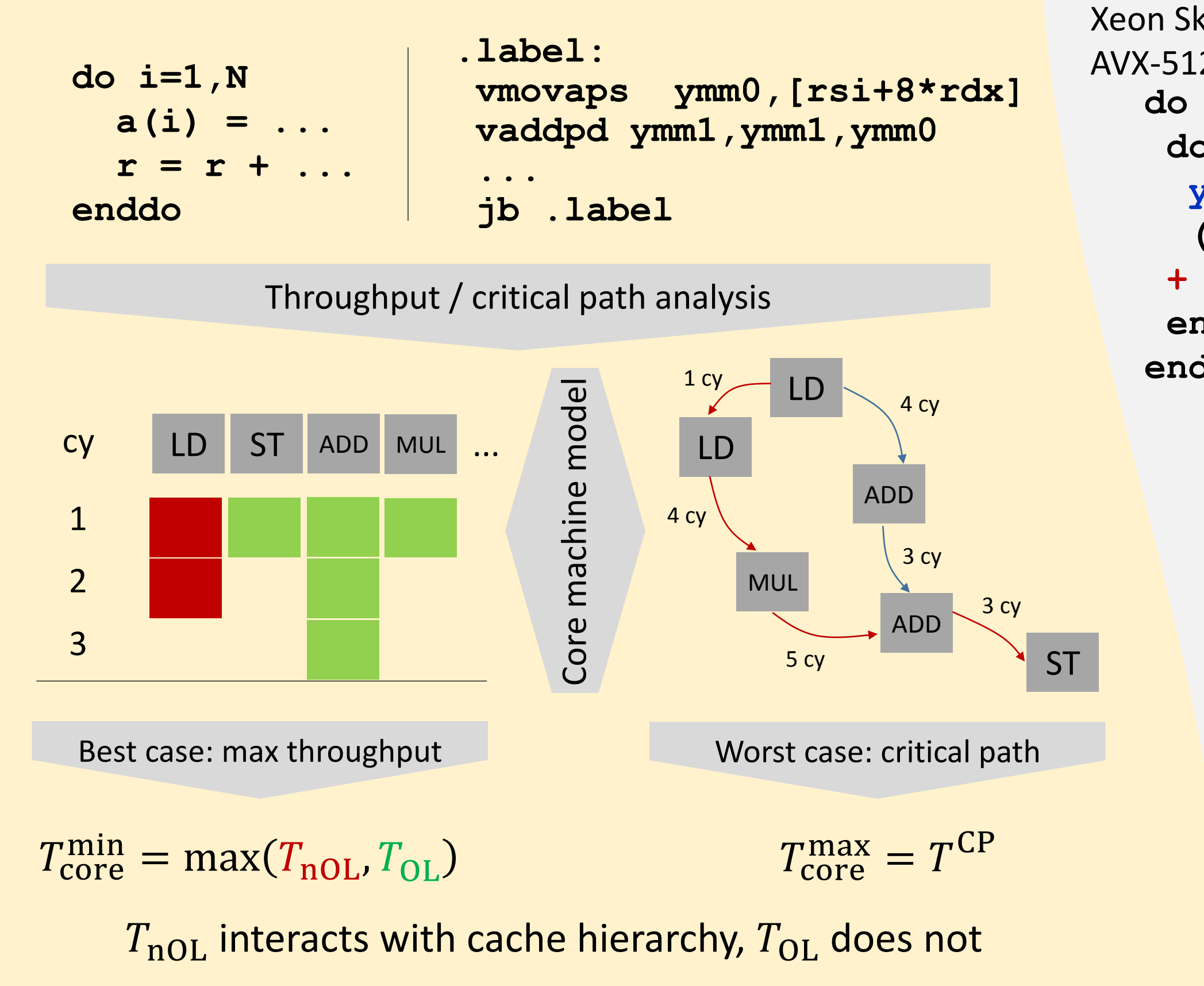
An analytic runtime model for loop code on the core and chip level



Cache architectures and transfer times



In-core execution model



Example: 2D DP 5-pt stencil on Xeon Skylake Gold 6148, AVX-512

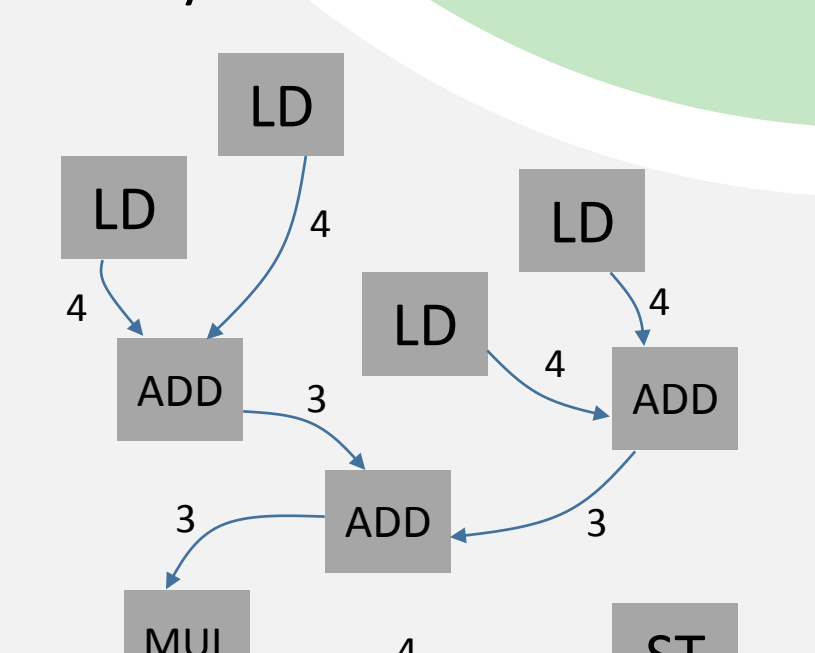
```
do j=1,N
do i=1,N
y(i,j) = 0.25d0 *
(x(i-1,j) + x(i+1,j)
+ x(i,j-1) + x(i,j+1))
enddo
enddo
```

Throughput analysis

	#uops	#cy
LD	4	2
ST	1	1
ADD	3	1.5
MUL	1	0.5
Σ	9	2

Retirement limit!
 $T_{nOL} = 2 \text{ cy}$
 $T_{OL} = 1.5 + 0.25 \text{ cy}$

CP analysis



$T_{CP} = 15 \text{ cy}$

Analysis tools:
• Pen & paper
• Intel Architecture Code Analyzer (IACA)
• Open-Source Architecture Code Analyzer (OSACA)
• Kerncraft

Putting the model together: Overlap assumptions

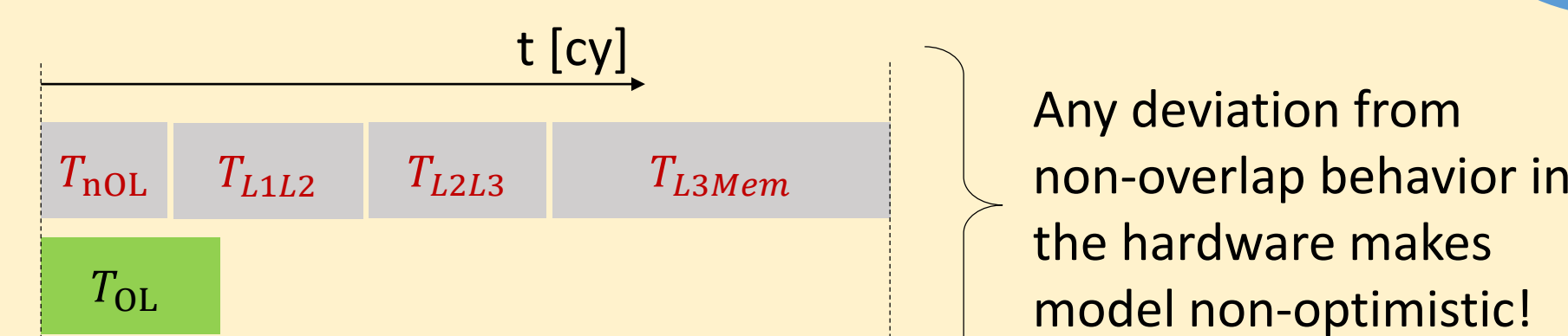
Notation for model contributions:

$\{T_{OL} || T_{nOL} | T_{L1L2} | T_{L2L3} | T_{L3Mem}\} = \{7 || 2 | 4 | 8 | 18.4\} \text{ cy/8 iter}$

How to put the contributions together? Overlap assumptions! (Part of the machine model)

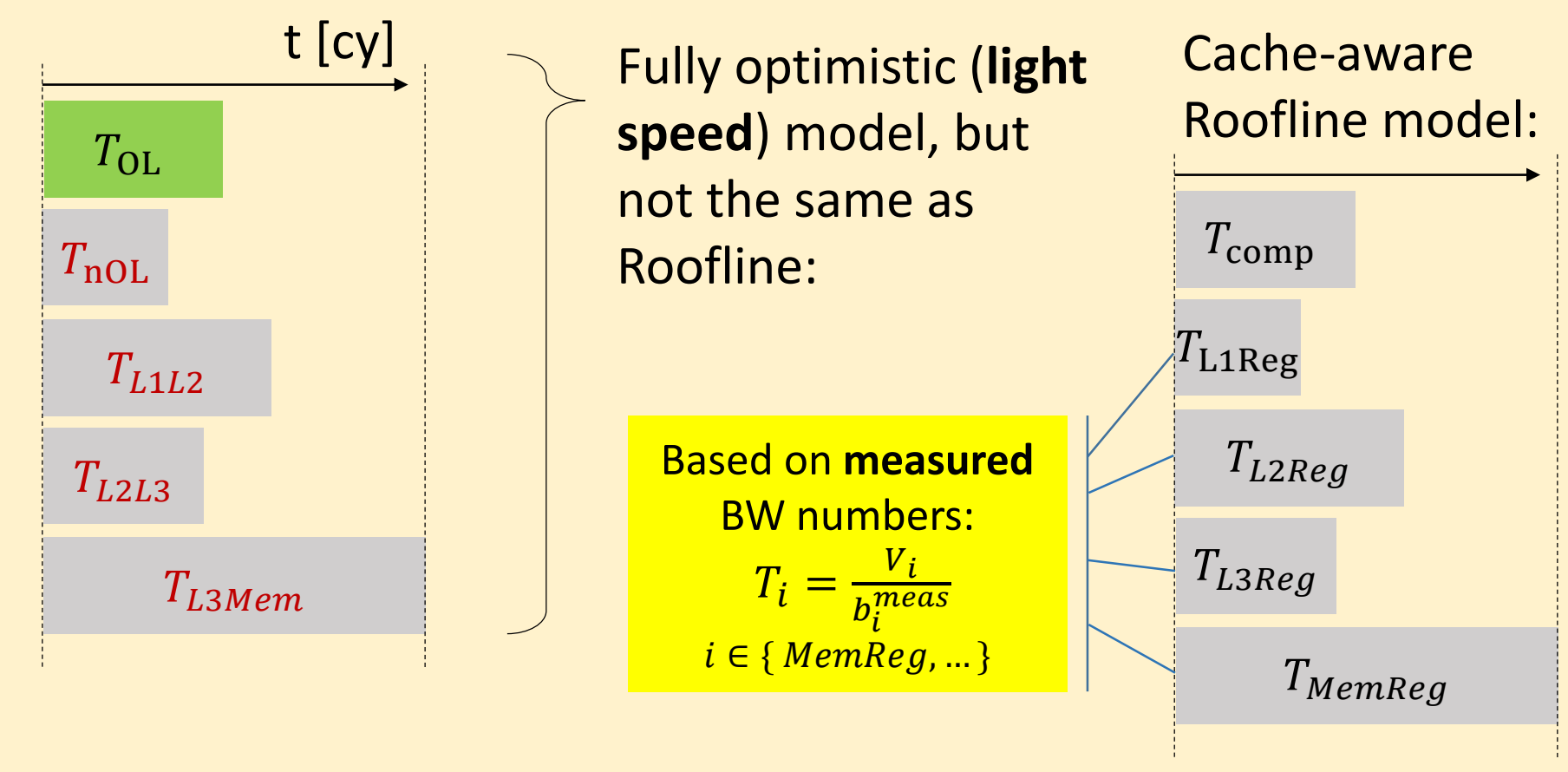
Most pessimistic assumption: **no overlap** of data-related contributions

$$T_{ECM}^{Mem} = \max(T_{OL}, T_{nOL} + T_{L1L2} + T_{L2L3} + T_{L3Mem})$$



Most optimistic assumption: **full overlap** of data-related contributions

$$T_{ECM}^{Mem} = \max(T_{OL}, T_{nOL}, T_{L1L2}, T_{L2L3}, T_{L3Mem})$$

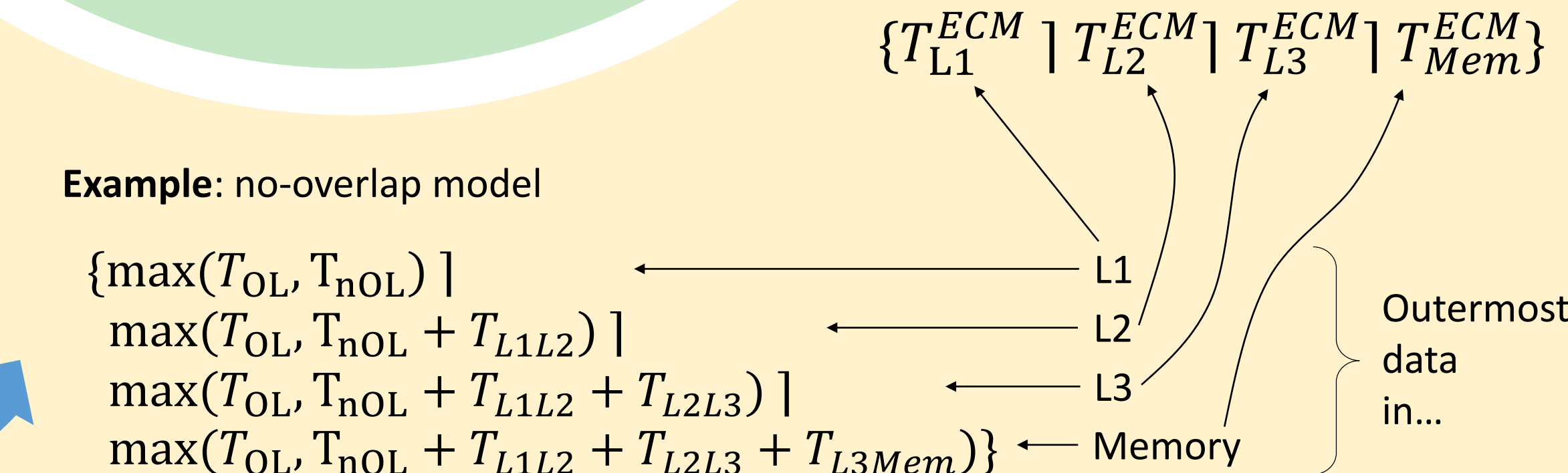


Mixed model: **partial overlap** of data-related contributions

Example: no overlap at L1, full overlap of all other contributions

$$T_{ECM}^{Mem} = \max(T_{OL}, T_{nOL} + T_{L1L2}, T_{L2L3}, T_{L3Mem})$$

Notation for model predictions



Example: AVX DP sum reduction w/ single accumulator on Broadwell EP CoD 2.3 GHz ($b_{L3Mem} = 32 \text{ GB/s}$)

$\{6 || 1 | 1 | 2 | 4.6\} \text{ cy/8 iter}$
 $\{6 | 6 | 6 | 7.6\} \text{ cy/8 iter}$

→ Constant performance up to L3

Multicore scaling and saturation

Optimistic assumption: Performance scaling is linear until a bandwidth bottleneck (e.g., b_{Mem}) is saturated → memory transfer time as lower limit

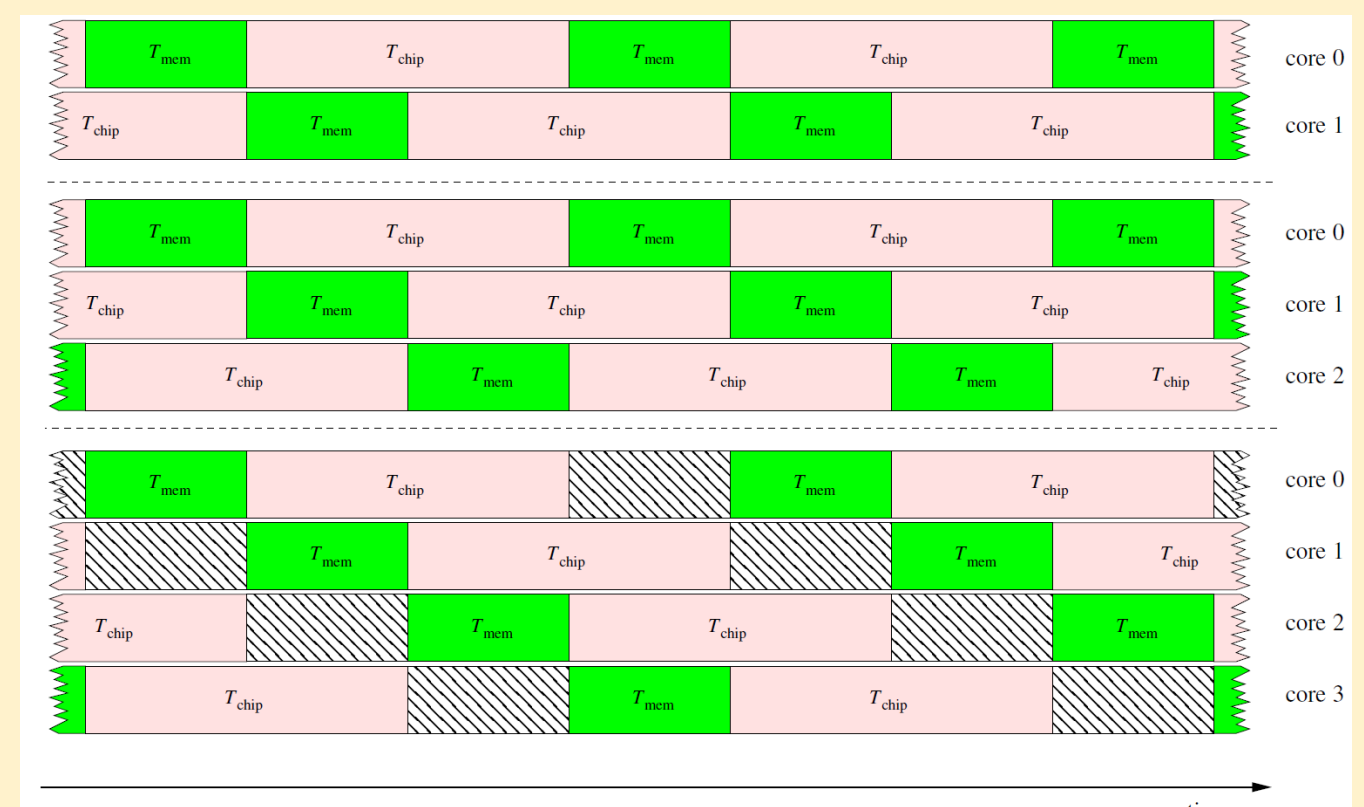
Runtime vs. cores (memory bottleneck):

$$T_{ECM}(n) = \max\left(\frac{T_{ECM}^{Mem}}{n}, T_{L3Mem}\right) \Rightarrow n_S = \left\lceil \frac{T_{ECM}^{Mem}}{T_{L3Mem}} \right\rceil$$

Example: AVX DP sum reduction from above

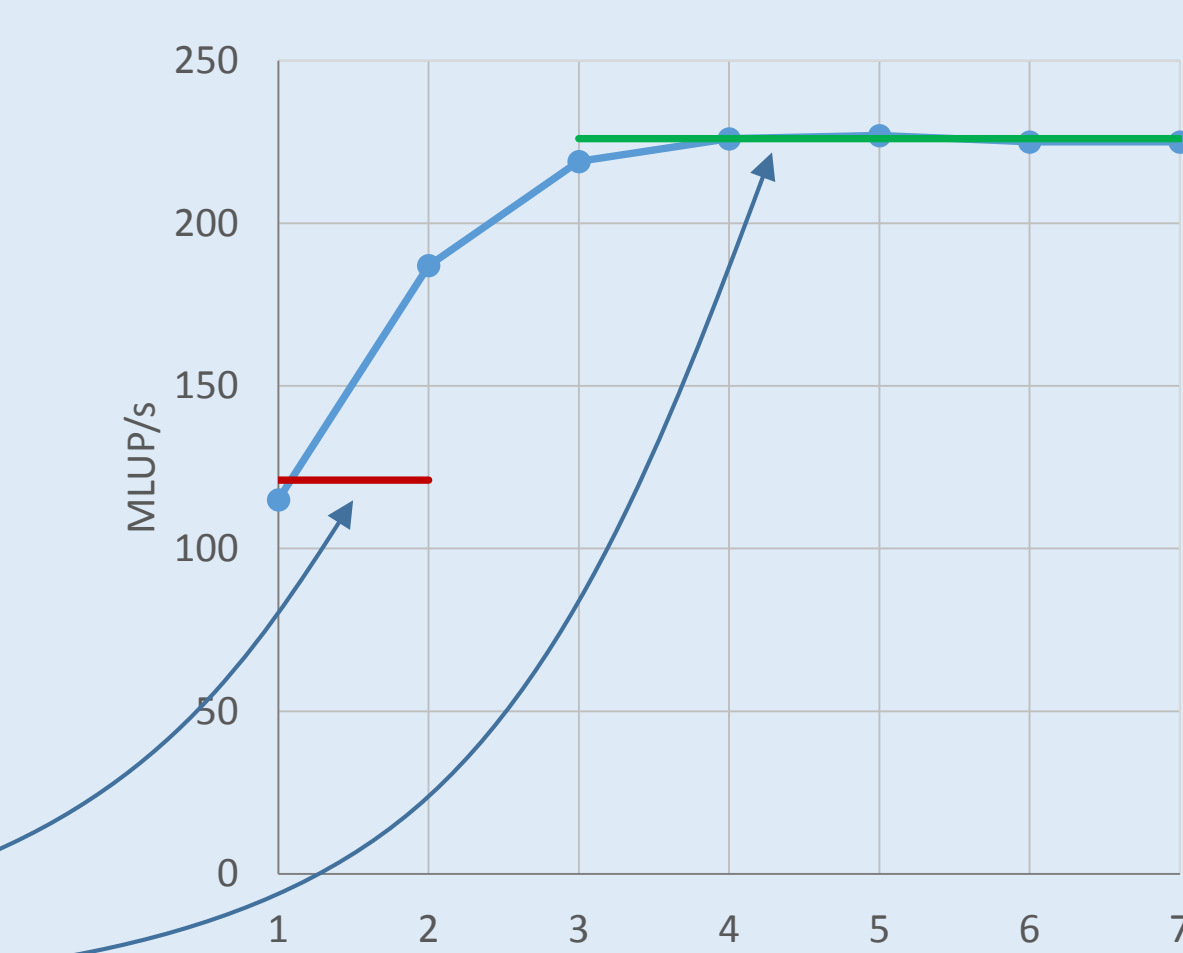
$\{6 || 1 | 1 | 2 | 4.6\} \text{ cy/8 iter}$
 $\{6 | 6 | 6 | 7.6\} \text{ cy/8 iter}$

$$\Rightarrow n_S = \left\lceil \frac{7.6}{4.6} \right\rceil = 2$$



Application: Conjugate Gradient solver

	ECM [cy/8 iter]	T_{ECM}^{Mem} [cy/8 iter]	n_S [cores]	Full domain limit [cy/8 iter]
$\vec{v} = A\vec{p}$	{ 8 4 6.7 10 16.9 }	37.6	3	16.9
$\lambda = \alpha_0 / \langle \vec{v}, \vec{p} \rangle$	{ 2 2 2.7 4 9.1 }	17.8	2	9.11
$\vec{x} = \vec{x} + \lambda \vec{p}$	{ 2 4 6 16.9 }	29.0	2	16.9
$\vec{r} = \vec{r} - \lambda \vec{v}$	{ 2 4 6 16.9 }	29.0	2	16.9
$\alpha_1 = \langle \vec{r}, \vec{p} \rangle$	{ 2 2 1.3 2 4.6 }	9.90	3	4.56
$\vec{p} = \vec{r} + \frac{\alpha_1}{\alpha_0} \vec{p}$, $\alpha_0 = \alpha_1$	{ 2 4 6 16.9 }	29.0	2	16.9
Sum		152		81.3



- Multi-loop code well represented
- Single core performance predicted with 5% error
- Saturated performance predicted with < 0.5% error
- Saturation point predicted approximately
- Future work: Include GS preconditioner → HPCG modeling

Overlap assumptions for current architectures

Intel Skylake SP: no overlap in data transfers, victim L3
 $T_{ECM}^{Mem} = \max(T_{OL}, T_{nOL} + T_{L1L2} + T_{L2L3} + T_{L3Mem})$

Intel Xeon ≤ Broadwell EP: no overlap

$$T_{ECM}^{Mem} = \max(T_{OL}, T_{nOL} + T_{L1L2} + T_{L2L3} + T_{L3Mem})$$

IBM Power8: overlap at L1

$$T_{ECM}^{Mem} = \max(T_{OL}, T_{nOL}, T_{L1L2}, T_{L2L3} + T_{L2Mem} + T_{L3Mem})$$

AMD Zen (Epyc): full overlap

$$T_{ECM}^{Mem} = \max(T_{nOL}, T_{L1L2}, T_{L2L3}, T_{L3Mem}, T_{OL})$$

Application: Blue Brain Project kernels

What is the computational cost of a synapse?

→ Study performance of brain simulation code via mini-apps

→ SSE4.2 vectorization by compiler

Case 1: Synaptic current kernel

```
for(_iml = 0; _iml < _cntml; ++_iml) {
_nd_idx = _ni[_iml]; v = vec_v[_nd_idx];
mggate[_iml] = 1.0 / ( 1.0 + exp( -0.062 * v ) * (mg[_iml]/3.57) );
g_AMPA[_iml] = gmax * ( B_AMPA[_iml] - A_AMPA[_iml] );
g_NMDA[_iml] = gmax * ( B_NMDA[_iml] - A_NMDA[_iml] ) * mggate[_iml];
i_AMPA[_iml] = g_AMPA[_iml] * ( v - e[_iml] );
i_NMDA[_iml] = g_NMDA[_iml] * ( v - e[_iml] );
i[_iml] = i_AMPA[_iml] + i_NMDA[_iml];
_g[_iml] = g_AMPA[_iml] + g_NMDA[_iml];
_rhs[_iml] = i[_iml]; _mfact = 1.e2 / ( _nd_area[_nd_area_indices[_iml]] );
_g[_iml] *= _mfact; _rhs[_iml] *= _mfact;
_vec_shadow_rhs[_iml] = _rhs[_iml]; _vec_shadow_d[_iml] = _g[_iml];
}
```

Challenges: **exp()** function call, some **divides**, some **indirect accesses**, integer register spill (many streams, call ABI restricts untouched registers)

- Streaming kernel, adjust data volume via knowledge about structure of index arrays (4% correction)
- Strong intra-iteration dependency chain
- exp()** overhead (thruput/latency) measured by microbenchmark, added to ECM

	"Ivy Bridge" E5-2660v2	"Haswell" E5-2695v3
Throughput ass.	{ 32.5 9.5 6.5 6.5 11.5 } → $T_{ECM}^{Mem} = 34 \text{ cy/iter}$	$T_{ECM}^{Mem} = 38.9 \text{ cy/iter}$
CP ass.	{ 49 9.5 6.5 6.5 11.5 } → $T_{ECM}^{Mem} = 49 \text{ cy/iter}$	$T_{ECM}^{Mem} = 50 \text{ cy/iter}$
Measured	48.7 cy/iter	39.4 cy/iter

IVY close to CP prediction, HSW data bound! Still saturating @ 3-5 cores on both CPUs!

Case 2: Sodium ion channel (NaTs2_t state)

```
for(_iml = 0; _iml < _cntml; ++_iml) {
_nd_idx = _ni[_iml]; v = vec_v[_nd_idx];
mAlpha[_iml] = ( 0.182 * ( v + 32.0 ) ) /
( 1.0 - ( exp( - ( v + 32.0 ) / 6.0 ) ) );
mBeta[_iml] = ( 0.124 * ( v - 32.0 ) ) /
( 1.0 - ( exp( - ( v - 32.0 ) / 6.0 ) ) );
mInf[_iml] = mAlpha[_iml] / ( mAlpha[_iml] + mBeta[_iml] );
mTau[_iml] = ( 1.0 / ( mAlpha[_iml] + mBeta[_iml] ) ) / 2.95;
hAlpha[_iml] = ( - 0.015 * ( v + 60.0 ) ) /
( 1.0 - ( exp( ( v + 60.0 ) / 6.0 ) ) );
hBeta[_iml] = ( - 0.015 * ( v - 60.0 ) ) /
( 1.0 - ( exp( ( v - 60.0 ) / 6.0 ) ) );
hInf[_iml] = hAlpha[_iml] / ( hAlpha[_iml] + hBeta[_iml] );
hTau[_iml] = ( 1.0 / ( hAlpha[_iml] + hBeta[_iml] ) ) / 2.95;
m[_iml] = m[_iml] + ( 1. - exp(dt*( - 1.0 / mTau[_iml] )) ) *
( - ( mInf[_iml] / mTau[_iml] ) ) /
( - 1.0 / mTau[_iml] ) - m[_iml];
h[_iml] = h[_iml] +
( 1. - exp(dt*( - 1.0 / hTau[_iml] )) ) *
( - ( hInf[_iml] / hTau[_iml] ) ) /
( - 1.0 / hTau[_iml] ) - h[_iml];
}
```

Challenges: **exp()** & **divides** dominate, no strong dependency chain, small data volume

"Ivy Bridge" E5-2660v2: Full throughput T_{OL}
{ 140 || 9.25 | 5.15 | 5.15 | 9 } → $T_{ECM}^{Mem} \geq 140 \text{ cy}$

Measurement: 191 cy Saturation @ 16 cores!