ERLANGEN REGIONAL COMPUTING CENTER

The Execution-Cache-Memory (ECM) Performance Model

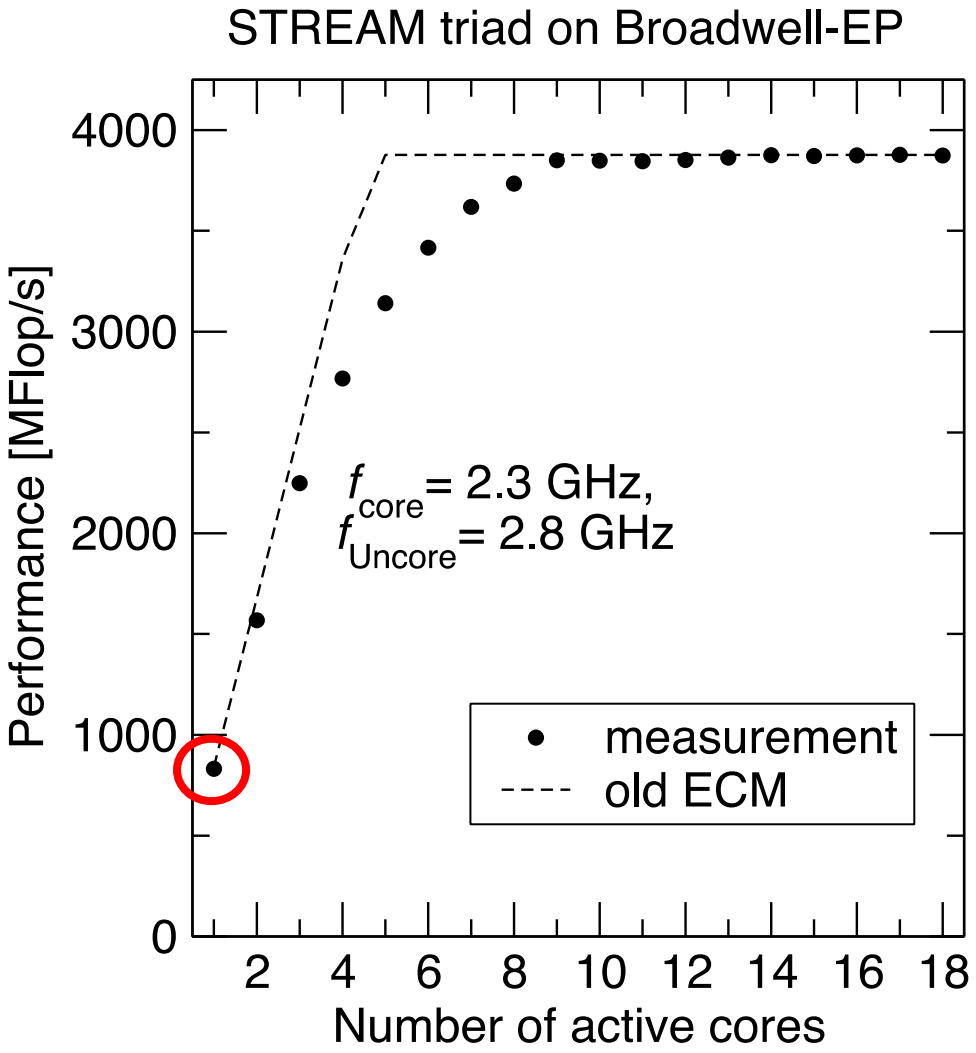
Georg Hager, Gerhard Wellein, Jan Eitzinger
Erlangen Regional Computing Center (RRZE)
Friedrich-Alexander-Universität Erlangen-Nürnberg

Intel Platform Performance Brown Bag
2018-10-25

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG

Motivation

Searching a good model for the single core performance of streaming loop kernels



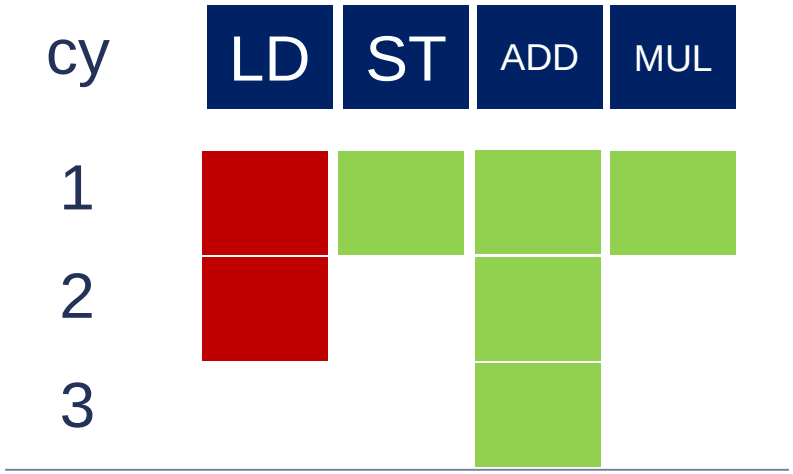
The ECM Model

ECM is a resource-based model for the runtime of loops on one core of a cache-based multicore CPU

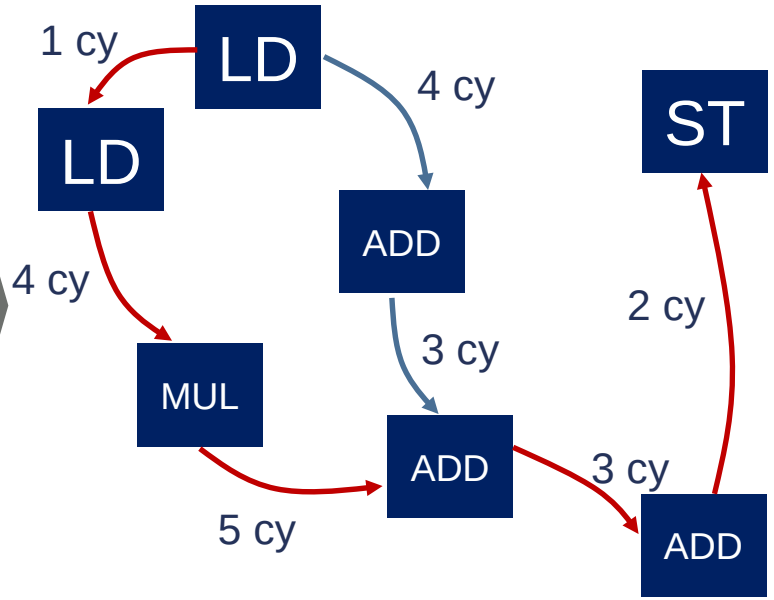
Major model assumptions:

- Steady-state loop code execution
  - No startup latencies, “infinitely long loop”
- No data access latencies
  - Can be added if need be
- Out-of-order scheduler works perfectly
  - But dependencies/critical paths can be taken into account

ECM model components: In-core execution



Core machine model



Best case: max throughput

Worst case: critical path

$$T_{\text{core}}^{\min} = \max(T_{\text{nOL}}, T_{\text{OL}})$$

$$T_{\text{core}}^{\max} = T^{\text{CP}}$$

T_{nOL} interacts with cache hierarchy, T_{OL} does not

ECM model components: Data transfer times

http://tiny.cc/kerncraft

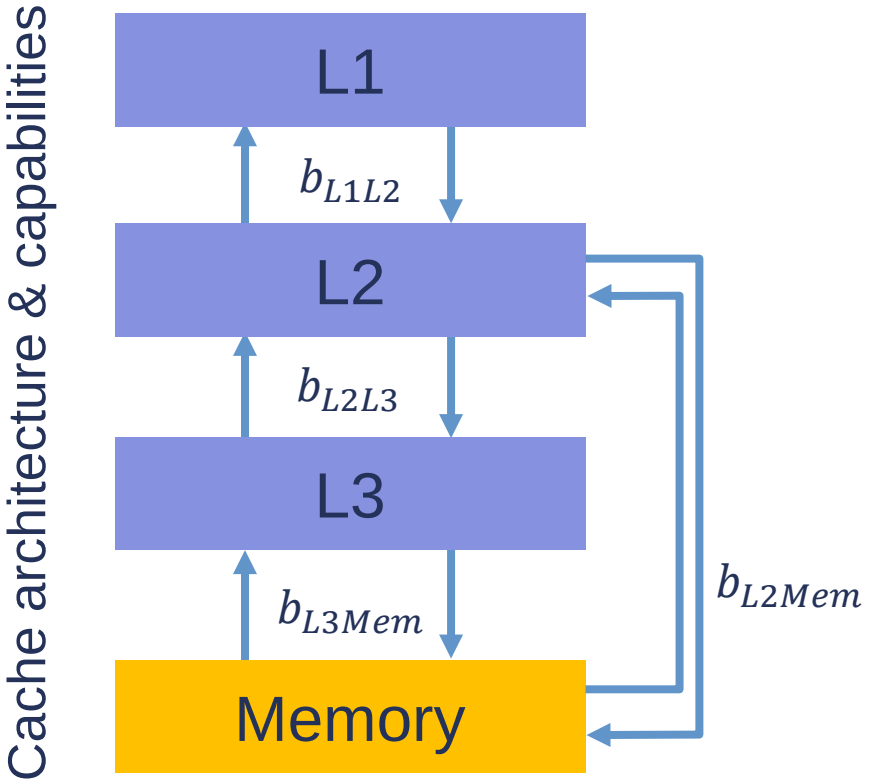
KERNCRAFT

Automatic Roofline/ECM modeling tool

- Optimistic transfer times through mem hierarchy
- $T_i = \frac{V_i}{b_i}$
- Transfer time notation for a given loop kernel:

$$\{T_{L1L2} | T_{L2L3} | T_{L3Mem}\} = \{ 4 \mid 8 \mid 18.4 \}\,\mathrm{cy/8\ iter}$$

- Input:
  - Cache properties (bandwidths, inclusive/exclusive)
  - Saturated memory bandwidth
  - Application data transfer prediction



ECM model components: Overlap assumptions (1)

- Notation for model contributions

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

- Most pessimistic overlap model: no overlap

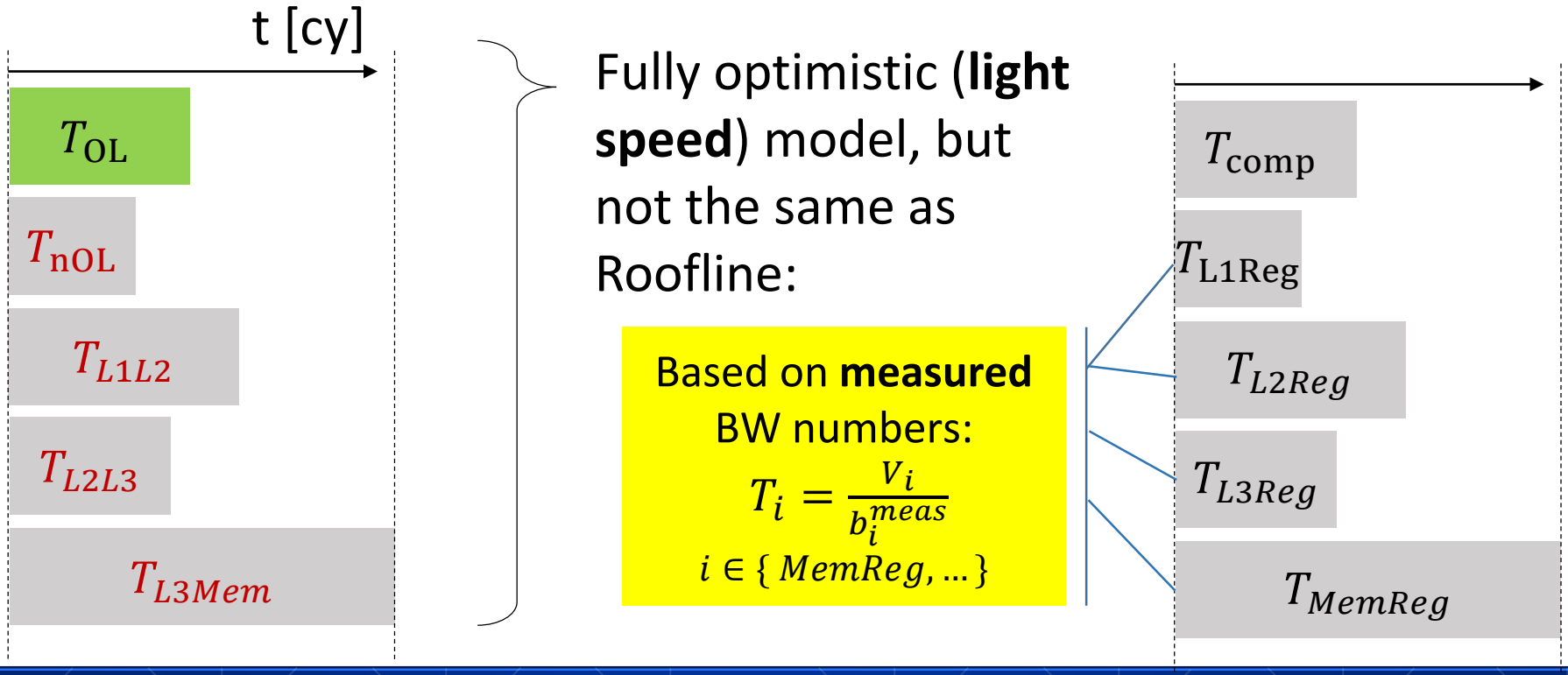
$$T_{ECM}^{Mem} = \max(T_{\mathrm{OL}}, T_{\mathrm{nOL}} + T_{L1L2} + T_{L2L3} + T_{L3Mem}) \text{ for in-mem data}$$

Appropriate for most Intel Xeon CPUs up to and including Broadwell

ECM model components: Overlap assumptions (2)

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

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

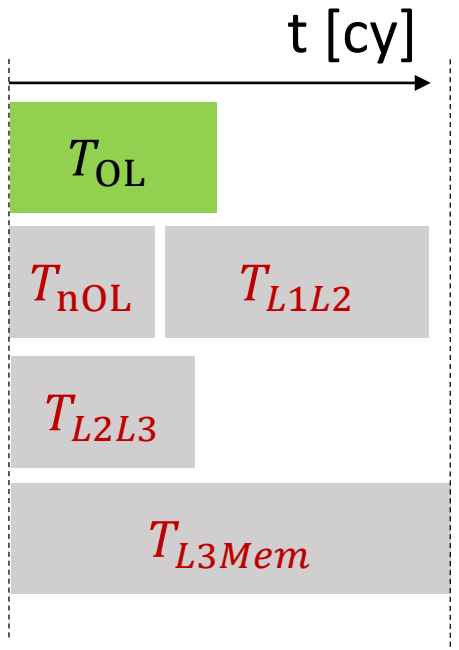


ECM model components: Overlap assumptions (3)

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

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

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



ECM model:
Notation for runtime predictions

$$\{T_{\mathrm{L1}}^{ECM} \rceil T_{L2}^{ECM} \rceil T_{L3}^{ECM} \rceil T_{Mem}^{ECM}\}$$

Example: no-overlap model

$$\{\max(T_{\mathrm{OL}}, \mathrm{T_{nOL}}) \rceil$$
$$\max(T_{\mathrm{OL}}, \mathrm{T_{nOL}} + T_{L1L2}) \rceil$$
$$\max(T_{\mathrm{OL}}, \mathrm{T_{nOL}} + T_{L1L2} + T_{L2L3}) \rceil$$
$$\max(T_{\mathrm{OL}}, \mathrm{T_{nOL}} + T_{L1L2} + T_{L2L3} + T_{L3Mem})\}$$

L1
L2
L3
Memory

data in...

ECM model: (Naive) saturation assuption

- Performance is assumed to scale across cores until a shared bandwidth bottleneck is hit

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

Roofline bandwidth ceiling

- This is (sometimes) too optimistic near the saturation point. For improvements see

J. Hofmann, G. Hager, and D. Fey: *On the accuracy and usefulness of analytic energy models for contemporary multicore processors*. Proc. ISC High Performance 2018. DOI: 10.1007/978-3-319-92040-5_2

2D 5-PT JACOBI STENCIL (DOUBLE PRECISION)

```
for(j=1; j < Nj-1; ++j)
 for(i=1; i < Ni-1; ++i)
  b[j][i] = (a[ j ][i-1] + a[ j ][i+1]
           + a[j-1][ i ] + a[j+1][ i ] ) * s;
```

Unit of work (1 CL): 8 LUPs

Data transfer per unit:

- 5 CL if layer condition violated in higher cache level
- 3 CL if layer condition satisfied

ECM Model for 2D Jacobi (AVX) on SNB 2.7 GHz

Radius-r stencil → (2r+1) layers have to fit

```
for(j=1; j < Nj-1; ++j)
 for(i=1; i < Ni-1; ++i)
  b[j][i] = (a[ j ][i-1] + a[ j ][i+1]
           + a[j-1][ i ] + a[j+1][ i ] ) * s;
```

Cache k has size C_k

Layer condition:

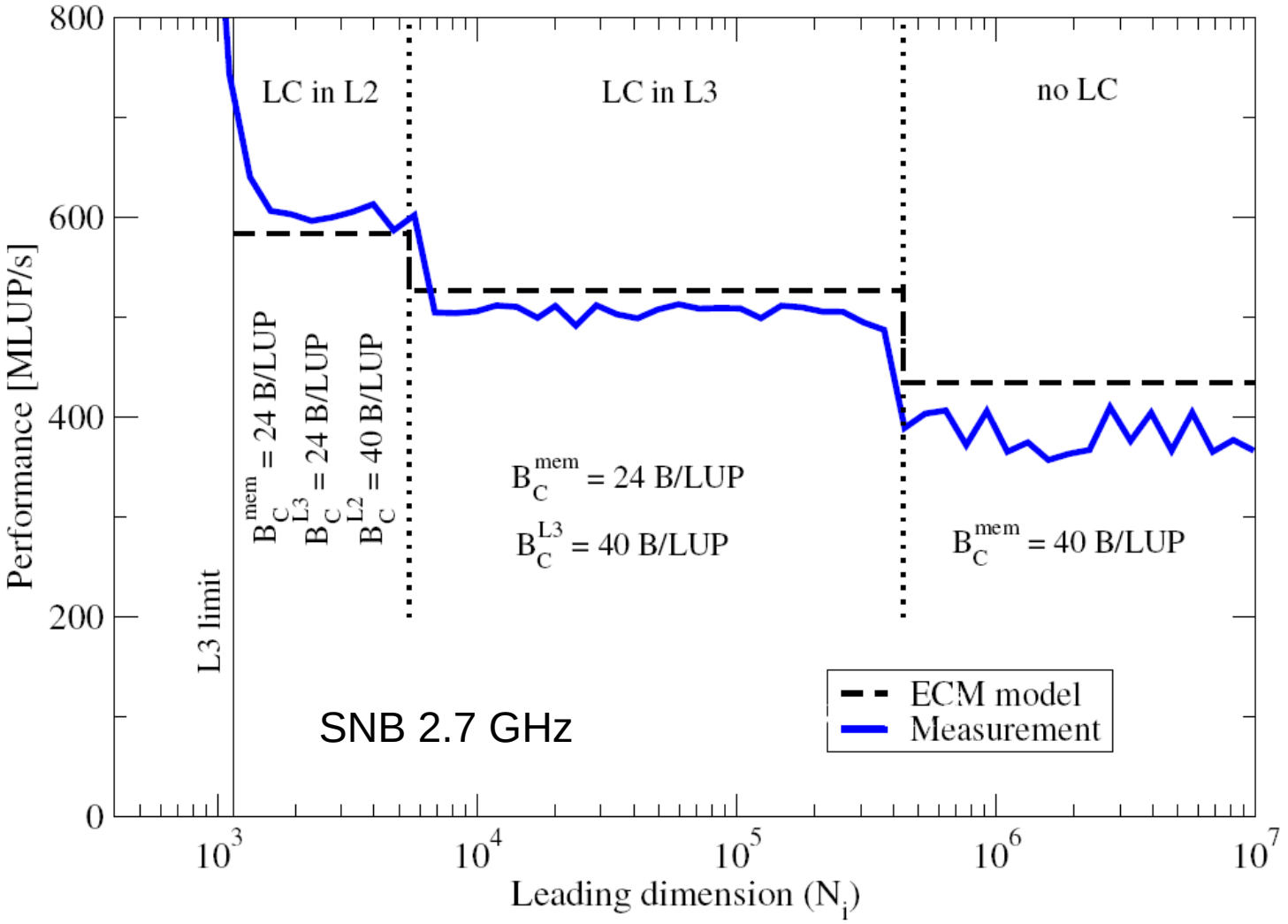
$$(2r+1) \cdot N_i \cdot 8\,\mathrm{B} < \frac{C_k}{2}$$

2D 5-pt: $r = 1$

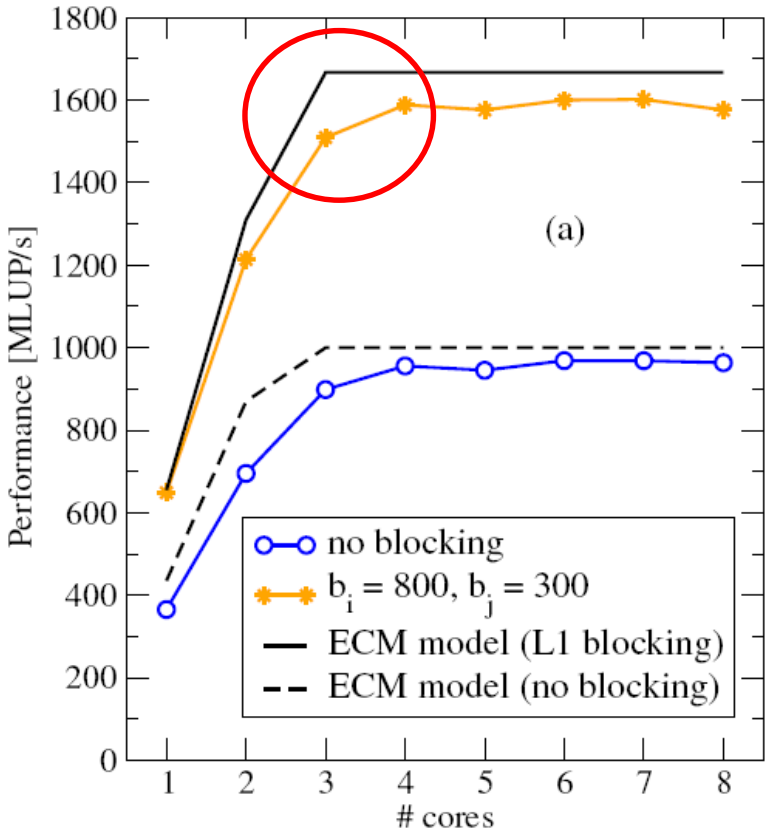
LC	ECM Model [cy]	prediction [cy]	$P^{\mathrm{mem}}_{\mathrm{ECM}}$ [MLUPS]	$N_i <$	n_S
L1	{6 ‖ 8 | 6 | 6 | 13}	{8 ⌉ 14 ⌉ 20 ⌉ 33}	659	683	3
L2	{6 ‖ 8 | 10 | 6 | 13}	{8 ⌉ 18 ⌉ 24 ⌉ 37}	587	5461	3
L3	{6 ‖ 8 | 10 | 10 | 13}	{8 ⌉ 18 ⌉ 28 ⌉ 41}	529	436900	4
—	{6 ‖ 8 | 10 | 10 | 22}	{8 ⌉ 18 ⌉ 28 ⌉ 50}	438	N/A	3

LC = layer condition satisfied in …

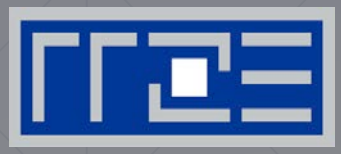
2D 5-pt serial in-memory performance and layer conditions



2D 5-pt multi-core scaling



A KERNEL FROM THE BLUE BRAIN PROJECT



A more complex situation

“Synaptic Current” kernel

- SSE4.2 vectorization
- Some indirect accesses, exp(), divides

```
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[area_indices[_iml]]);
  _g[_iml] *=  _mfact; _rhs[_iml] *= _mfact;
  _vec_shadow_rhs[_iml] = _rhs[_iml]; _vec_shadow_d[_iml] = _g[_iml];
}
```

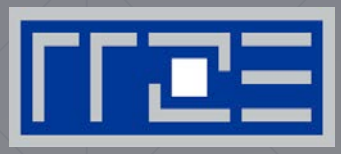
“Synaptic Current” kernel

	“Ivy Bridge” E5-2660v2	“Haswell” E5-2695v3
Throughput assumption	$\{ 32.5 \parallel 9.5 \mid 6.5 \mid 6.5 \mid 11.5 \}$ $T_{\text{Mem}}^{\text{ECM}} = 34$ cy/iter	$T_{\text{Mem}}^{\text{ECM}} = 38.9$ cy/iter
CP assumption	$\{ 49 \parallel 9.5 \mid 6.5 \mid 6.5 \mid 11.5 \}$ $T_{\text{Mem}}^{\text{ECM}} = 49$ cy/iter	$T_{\text{Mem}}^{\text{ECM}} = 50$ 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!

MODELING A CONJUGATE-GRADIENT SOLVER

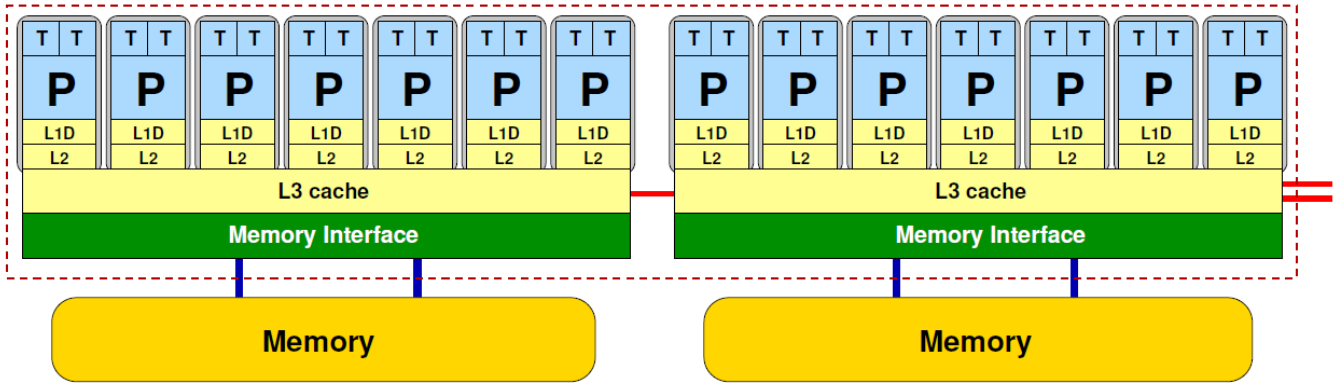


Building a model from components

A matrix-free CG solver

- 2D 5-pt FD Poisson problem
- Dirichlet BCs, matrix-free
- N_x x N_y = 40000 × 1000 grid

- CPU: **Haswell E5-2695v3 CoD mode**



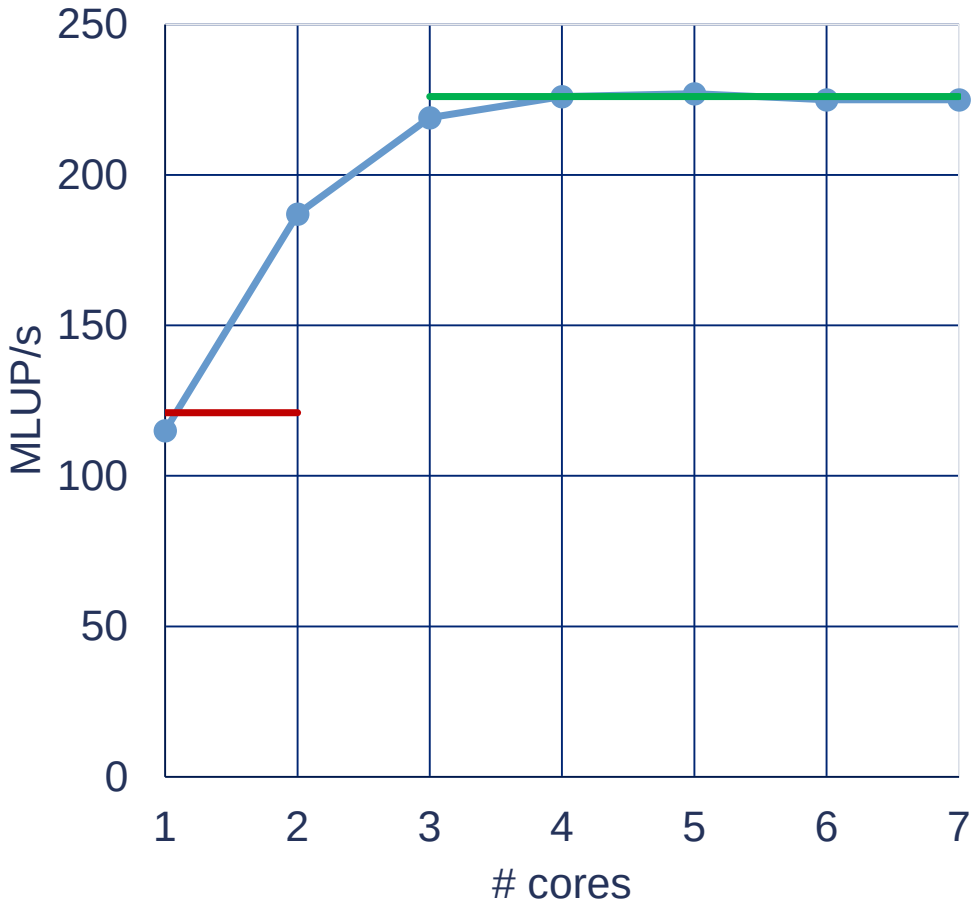
ECM model composition

Naive implementation of all kernels (omp parallel for)

while(α_0 < tol):	T_x [cy/8 iter]	T_{Mem}^{ECM} [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
$\alpha_1 = \langle \vec{r}, \vec{r} \rangle$	{ 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**

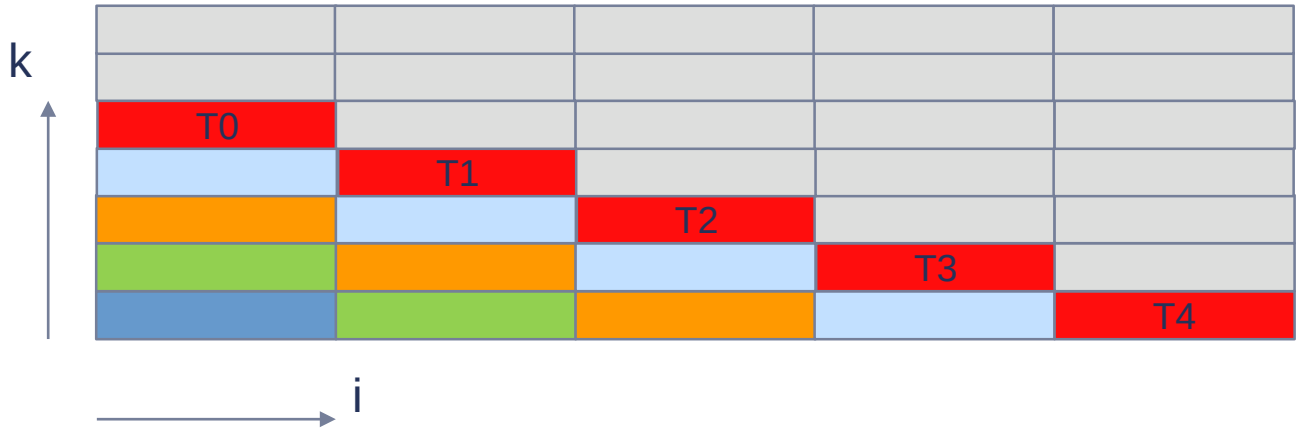
CG performance – 1 core to full socket

- Multi-loop code well represented
- Single core performance predicted with 5% error
- Saturated performance predicted with < 0.5% error
- Saturation point predicted approximately
  - Can be fixed by improved ECM model



CG with GS preconditioner: Naïve parallelization

Pipeline parallel processing: OpenMP barrier after each wavefront update (ugh!)



CG with GS preconditioner: additional kernels

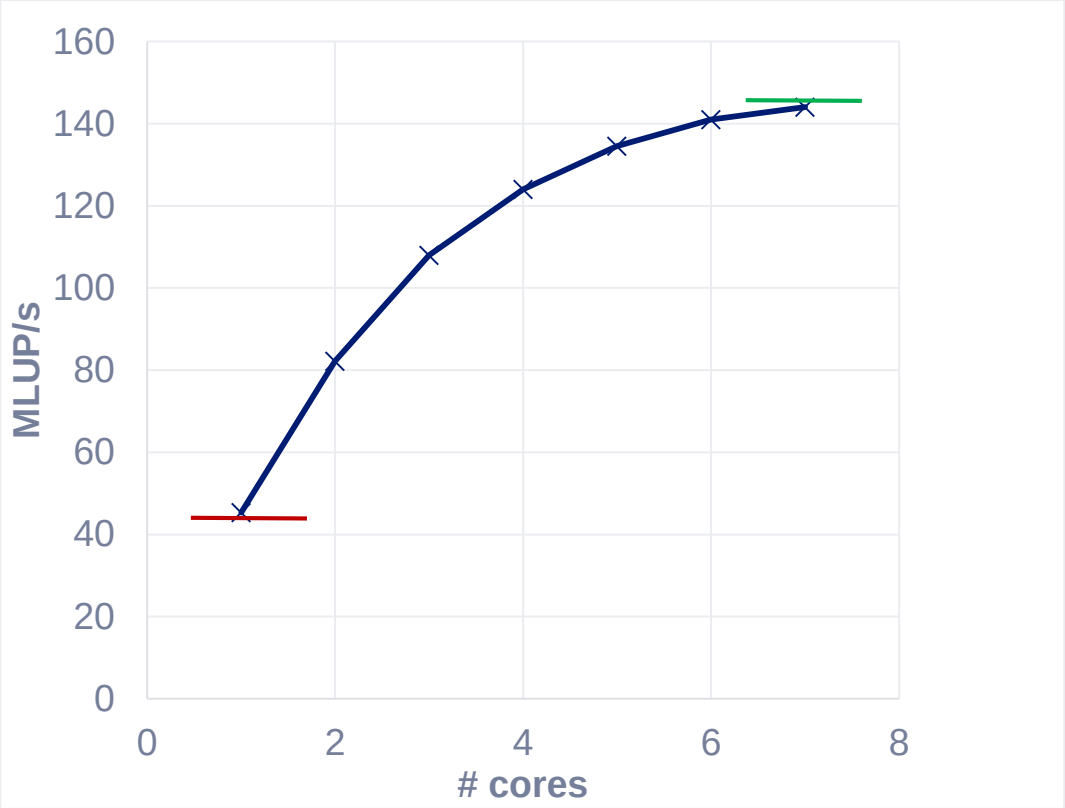
Intel IACA

	T_x [cy/8 iter]	T_{Mem}^{ECM} [cy/8 iter]	n_s [cores]	Full domain limit [cy/8 iter]
Non-PC model		**152**		**81.3**
$\vec{z} = P\vec{r}$ (fw)	{ 108 || 16 | 5.4 | 8 | 16.9 }	108	7	16.9
$\vec{z} = P\vec{r}$ (bw)	{ 138 || 16 | 4.0 | 6 | 11.3 }	138	13	19.7
$\alpha = \langle \vec{r}, \vec{z} \rangle$	{ 2 || 2 | 2.7 | 4 | 9.1 }	17.8	2	9.1
	Sum	**416**		**127**

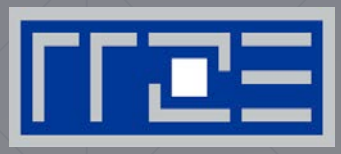
- Back substitution does not saturate the memory bandwidth!
  - → full algorithm does not fully saturate
- Impact of barrier still negligible overall, but noticeable in the preconditioner

PCG measurement

- <2% model error for single threaded and saturated performance
- Expected large impact of barrier at smaller problem sizes in x direction



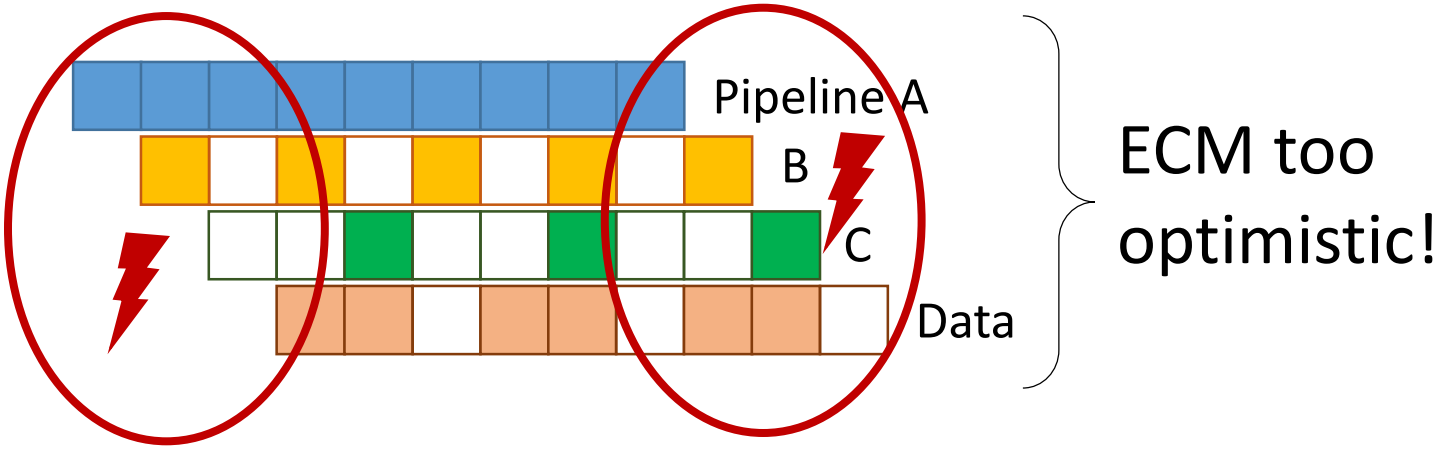
PROBLEMS AND OPEN QUESTIONS



What ECM cannot do (well)

Non-steady-state execution

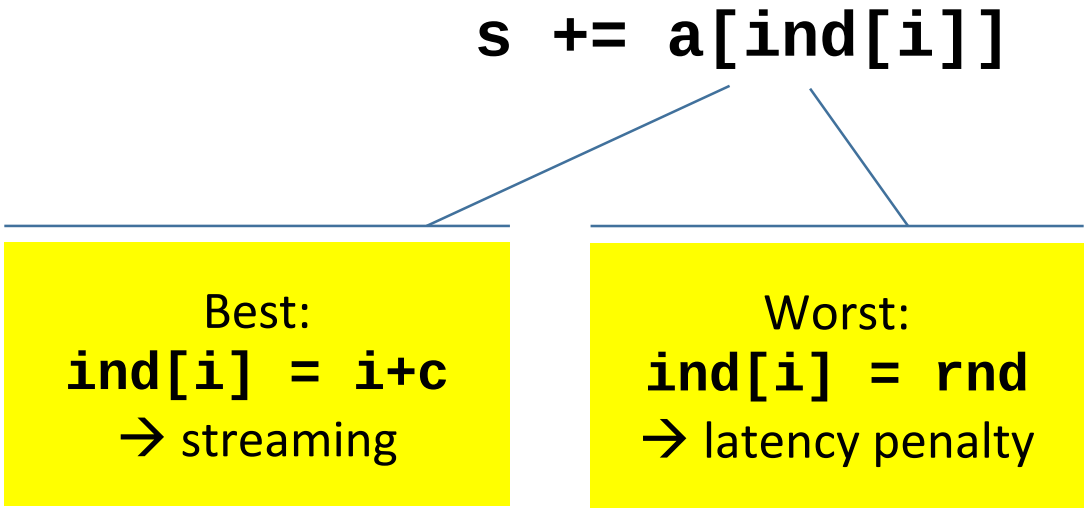
- Wind-up/wind-down effects are not part of the model



- May be added via corrections

Irregular data access

- Indirect != irregular



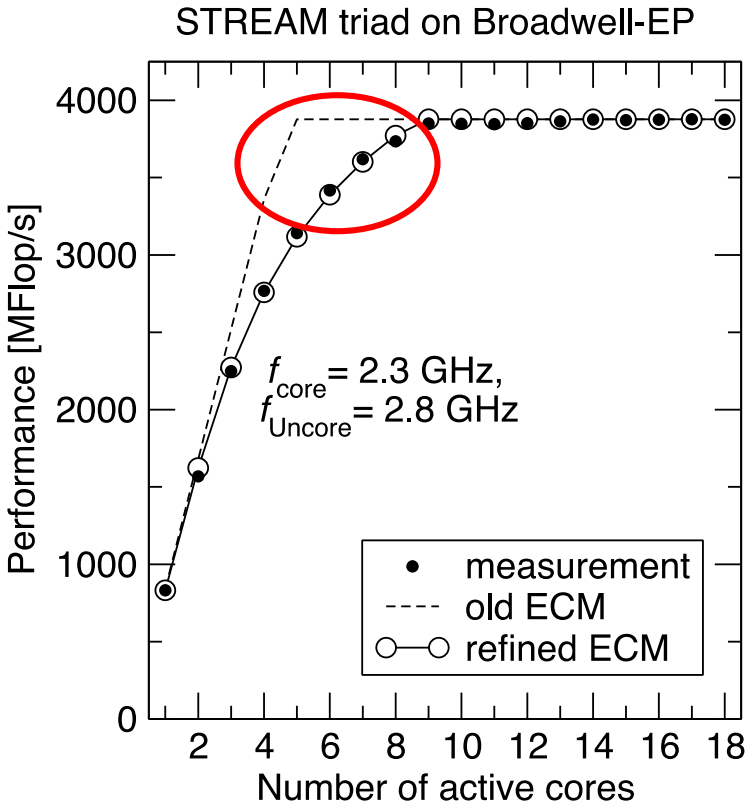
- Unknown access order → only best/worst-case analysis possible

Saturation

- Original ECM model too optimistic near saturation point
- Refinement: Adaptive latency penalty, depends on bus utilization $u(n)$:

$$u(1) = \frac{T_{L3Mem}}{T_{Mem}^{ECM}} \leftarrow \text{single-core model}$$

$$u(n) = \frac{T_{L3Mem}}{T_{Mem}^{ECM} + (n-1)u(n-1)p_0}$$



Fit parameter, not code independent → future work

Thank you.

https://hpc.fau.de/research/ecm