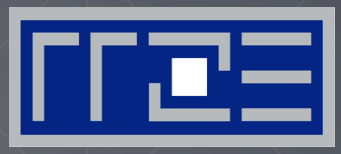
ERLANGEN REGIONAL COMPUTING CENTER



Making sense of temporally blocked stencil performance via analytic modeling

Georg Hager[1], Tareq Malas[2]

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

[2]National Energy Research Scientific Computing Center (NERSC), Berkeley, CA

7th AICS International Symposium

Emerging Numerical Techniques for Exascale and Post-Moore Era

February 23-24, 2016, Kobe, Japan

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG

Motivation

- Analytic performance modeling:

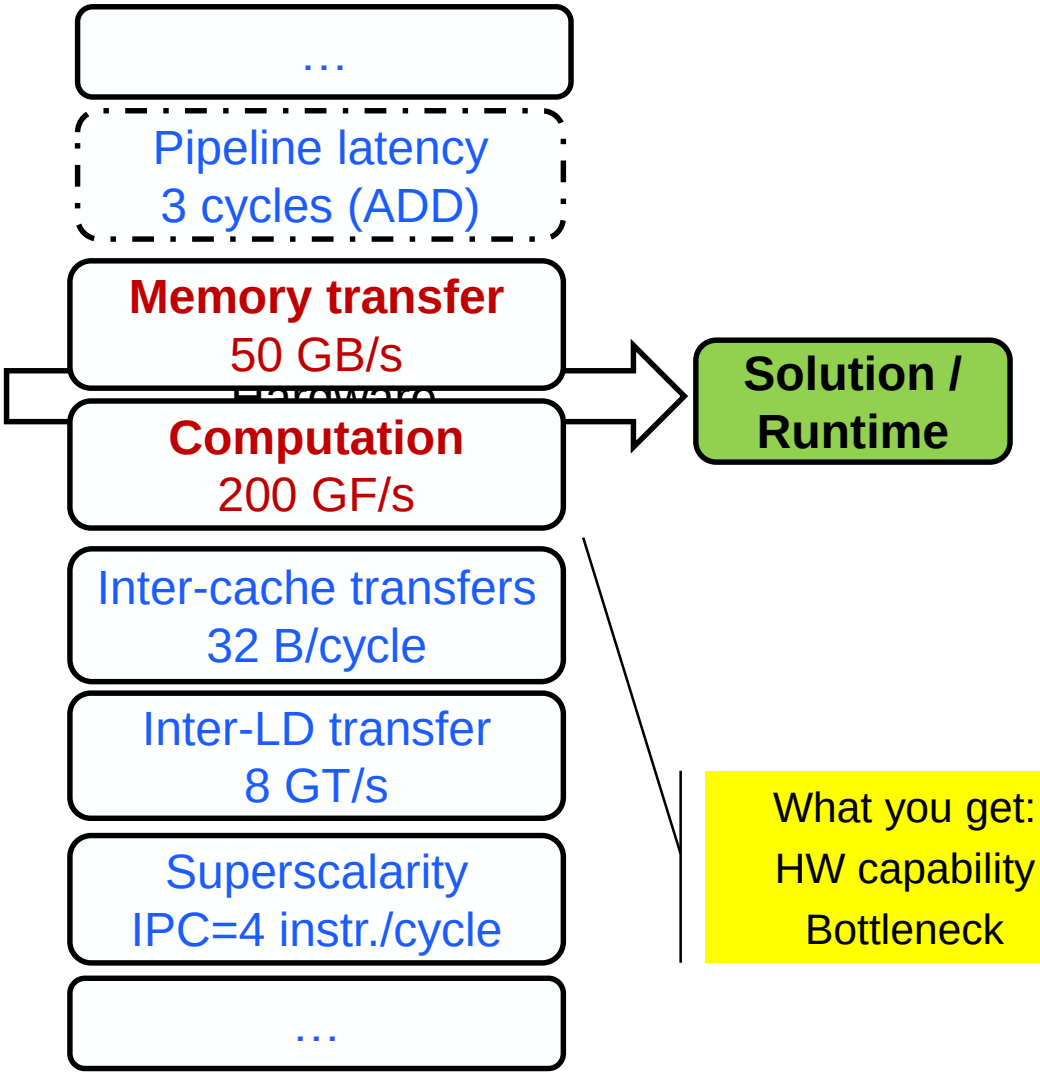
Constructing a simplified model for the interaction between software and hardware in order to understand lowest-order performance behavior

- Basic questions addressed by analytic performance models
  - What is the bottleneck?
  - What is the next bottleneck after optimization?
  - Impact of processor frequency and socket scalability → energy efficiency

- What if the model fails?
  - We learn something
  - We may still be able to use the model in a less predictive way

Potential hardware capabilities, a.k.a. bottlenecks

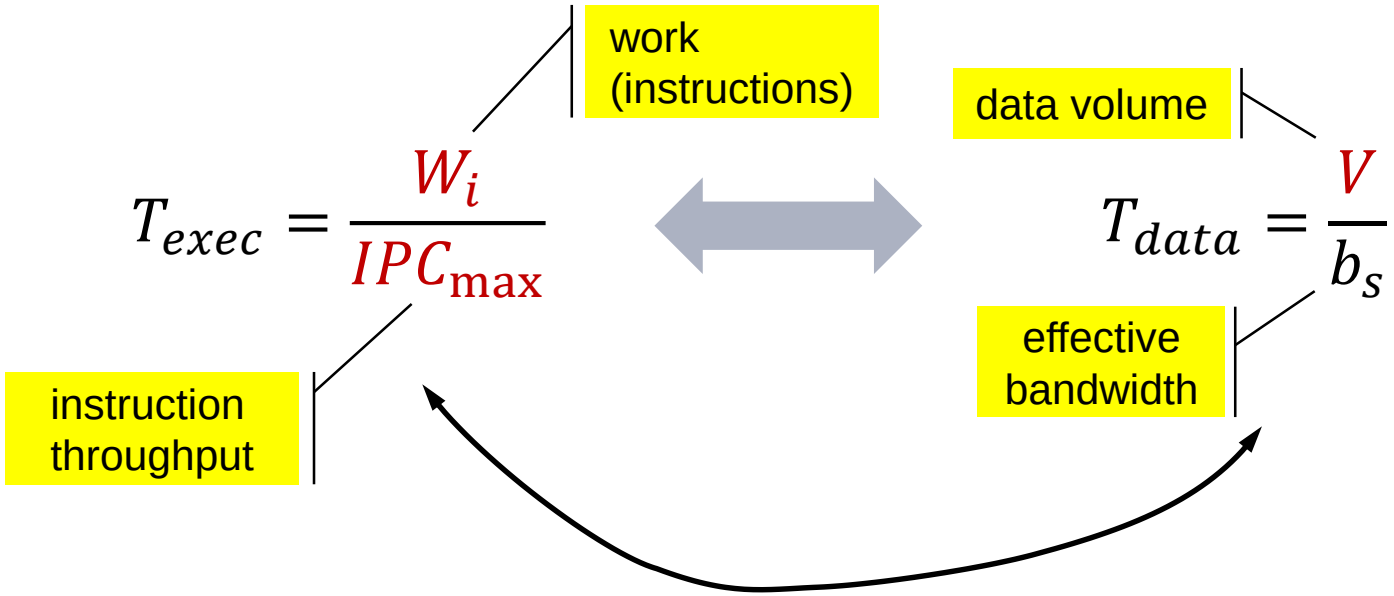
```
!$OMP PARALLEL DO
do j = 1 , 7000
 do i = 1 , 80000
   y(i,j) = b*(x(i-1,j)+
               x(i+1,j)+
               x(i,j-1)+
               x(i,j+1))
 enddo
enddo
!$OMP END PARALLEL DO
```



What you need:
Application requirements

What you get:
HW capability
Bottleneck

Start with the basics: Resources for program execution



Optimal ("balanced") co-design if we have equality

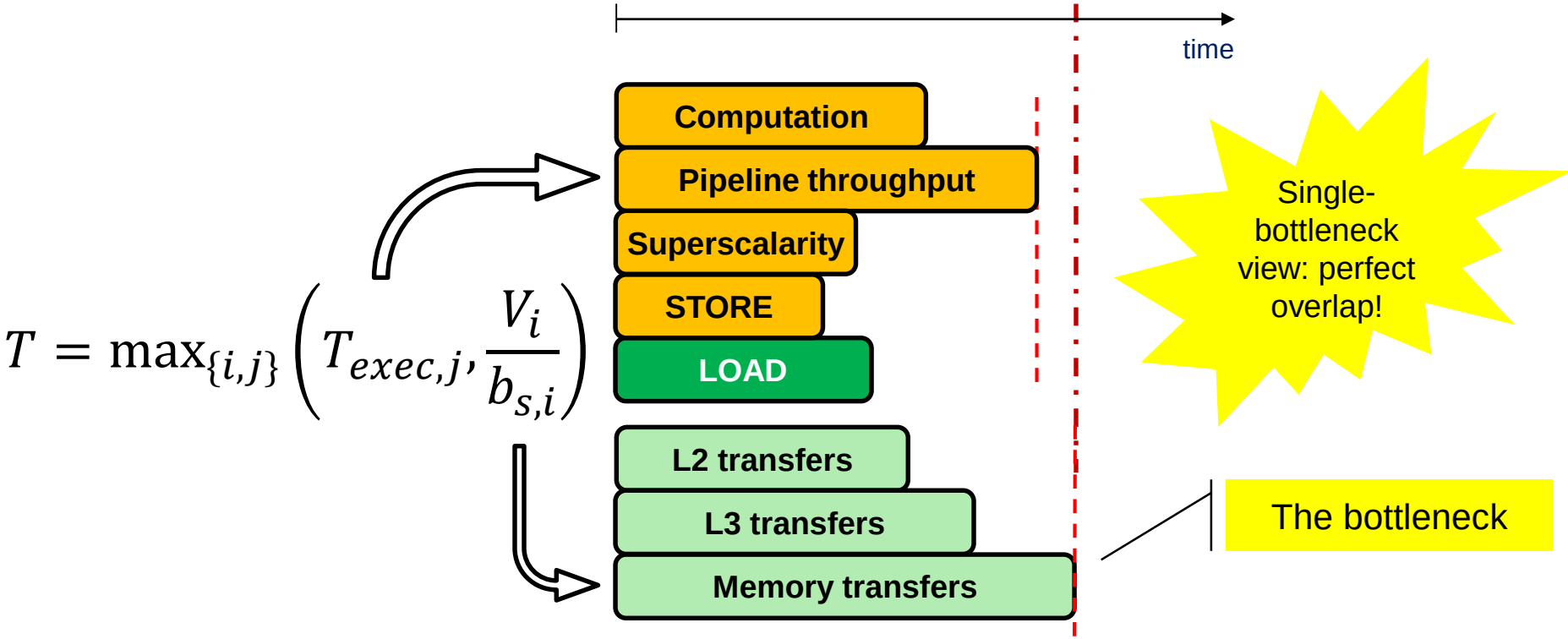
H.T. Kung: *Memory requirements for balanced computer architectures*. Proc. ISCA'86, DOI: 10.1145/17356.17362

R.W. Hockney and I.J. Curington: $f_{1/2}$: *A parameter to characterize memory and communication bottlenecks*. Parallel Computing 10, 277-286 (1989). DOI: 10.1016/0167-8191(89)90100-2

Lowest order model: Roofline model (RLM)

Williams, Waterman, Patterson (2009), DOI: 10.1145/1498765.1498785

Limited resources impose upper (lower) performance (runtime) limits



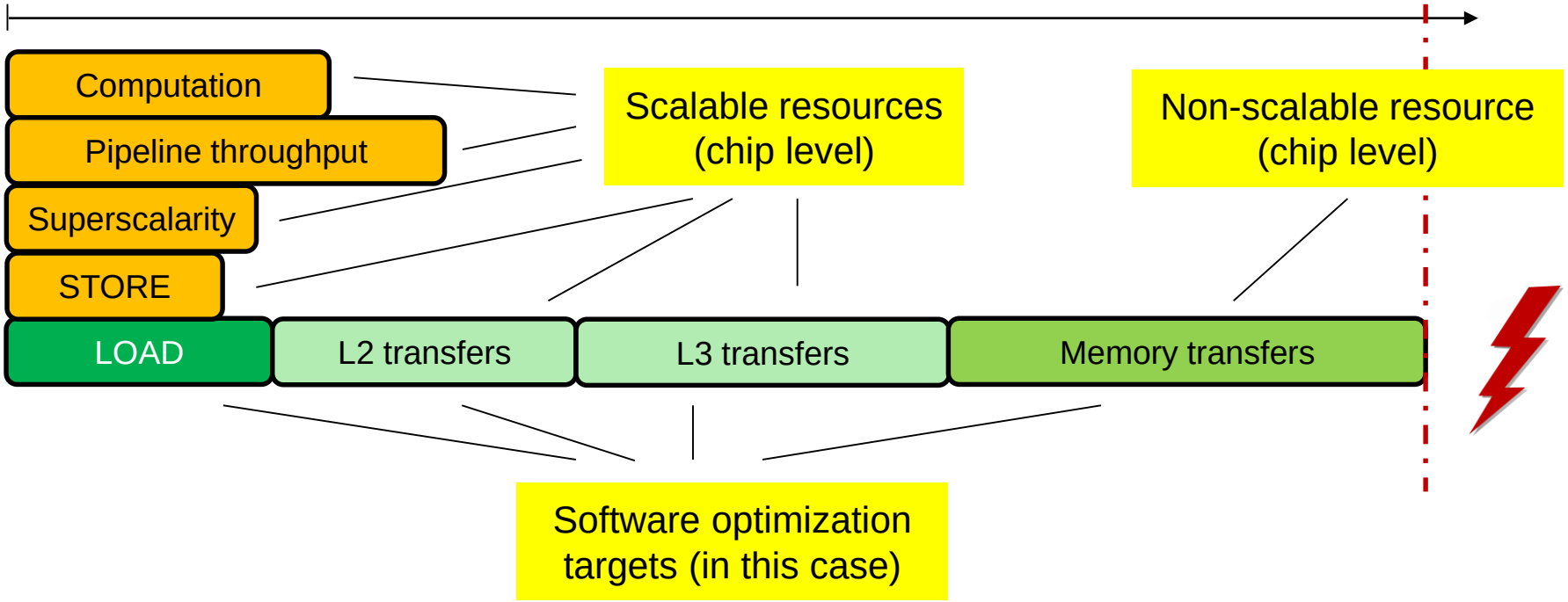
$$T = \max_{\{i,j\}} \left(T_{exec,j}, \frac{V_i}{b_{s,i}} \right)$$

Notation:

$$\{T_{OL} \parallel T_{nOL} \mid T_{L2} \mid T_{L3} \mid T_{Mem}\} \xrightarrow{prediction} \{\overbrace{\max(T_{OL}, T_{nOL})}^{\text{Data in L1}} \rceil \overbrace{\max(T_{OL}, T_{nOL}, T_{L2})}^{\text{Data in L2}} \rceil \ \ldots\}$$

Next to lowest order: Execution Cache Memory (ECM) Model

Simple bottleneck picture does not hold for non-overlap scenarios:
→ ECM single core model for Intel x86 architectures



$$\{T_{OL} \parallel T_{nOL} \mid T_{L1L2} \mid T_{L2L3} \mid T_{L3Mem}\} \xrightarrow{prediction} \{\max(T_{OL}, T_{nOL}) \rceil \max(T_{OL}, T_{nOL} + T_{L1L2}) \rceil \ \ldots\}$$

What about multiple cores?

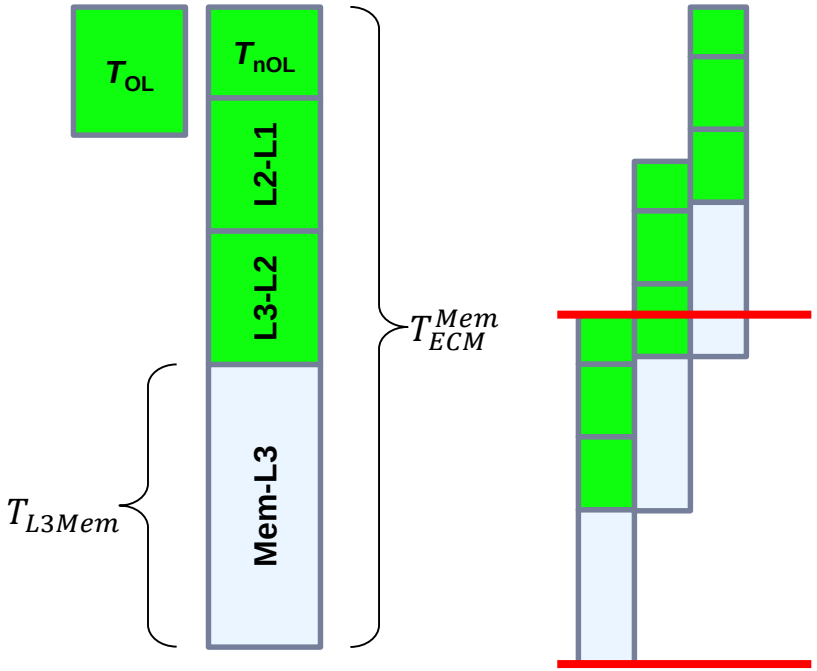
Main assumption: Performance scaling is linear until a bandwidth bottleneck (b_S) is hit

Performance vs. cores (Memory BN):

$$P(n) = \min\left(nP(1), \frac{b_S^{Mem}}{B_C^{Mem}}\right)$$

Number of cores at saturation:

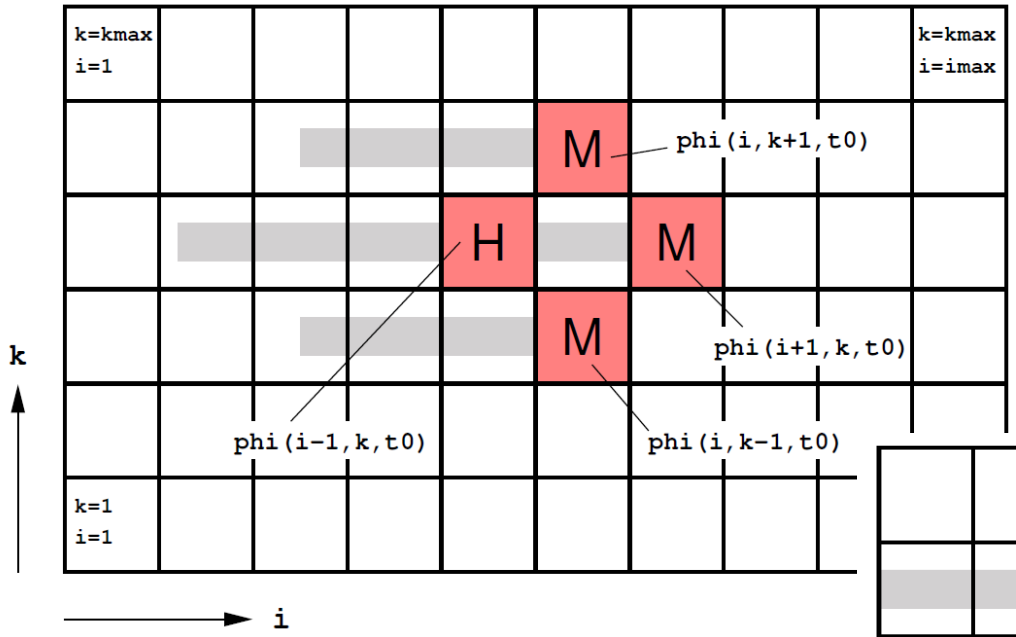
$$n_S = \left\lceil \frac{b_S/B_C}{P(1)} \right\rceil = \left\lceil \frac{T_{ECM}^{Mem}}{T_{L3Mem}} \right\rceil$$



Example:

$$\{ 8 \parallel 6 \mid 9 \mid 9 \mid 19 \} \text{ cy}, \qquad \{ 8 \rceil 15 \rceil 24 \rceil 43 \} \text{ cy} \implies n_S = \left\lceil \frac{43}{19} \right\rceil = 3$$

Example: 2D 5-point stencil & layer conditions

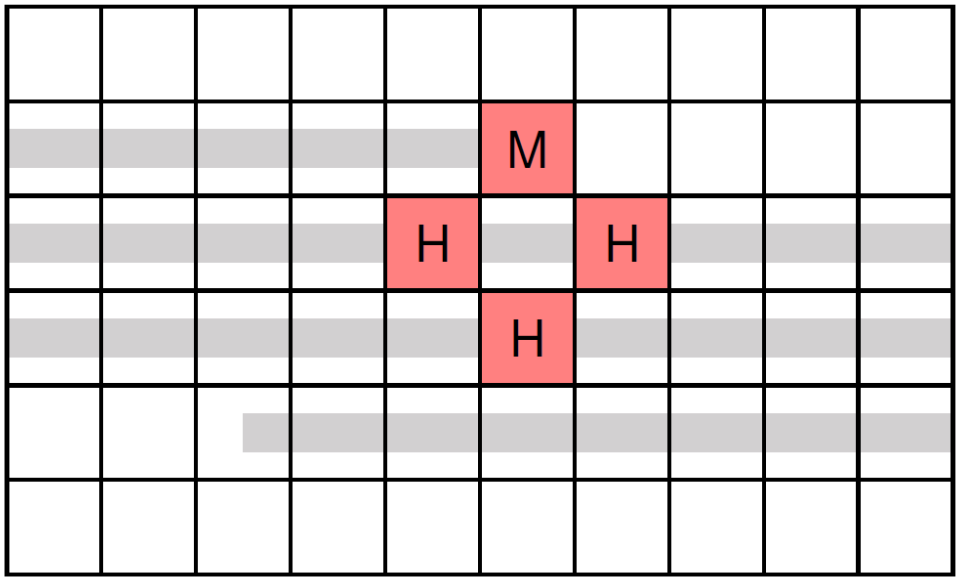


Cache k has size C_k

Layer condition (height r stencil):

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

2D 5-pt: $r = 1$



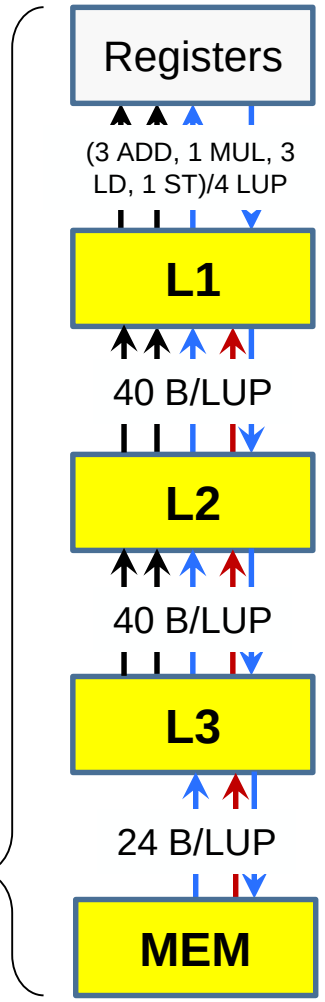
Predictive modeling: ECM Model for 2D 5-pt w/AVX on SNB 2.7 GHz

Radius-1 stencil → 3 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;
```

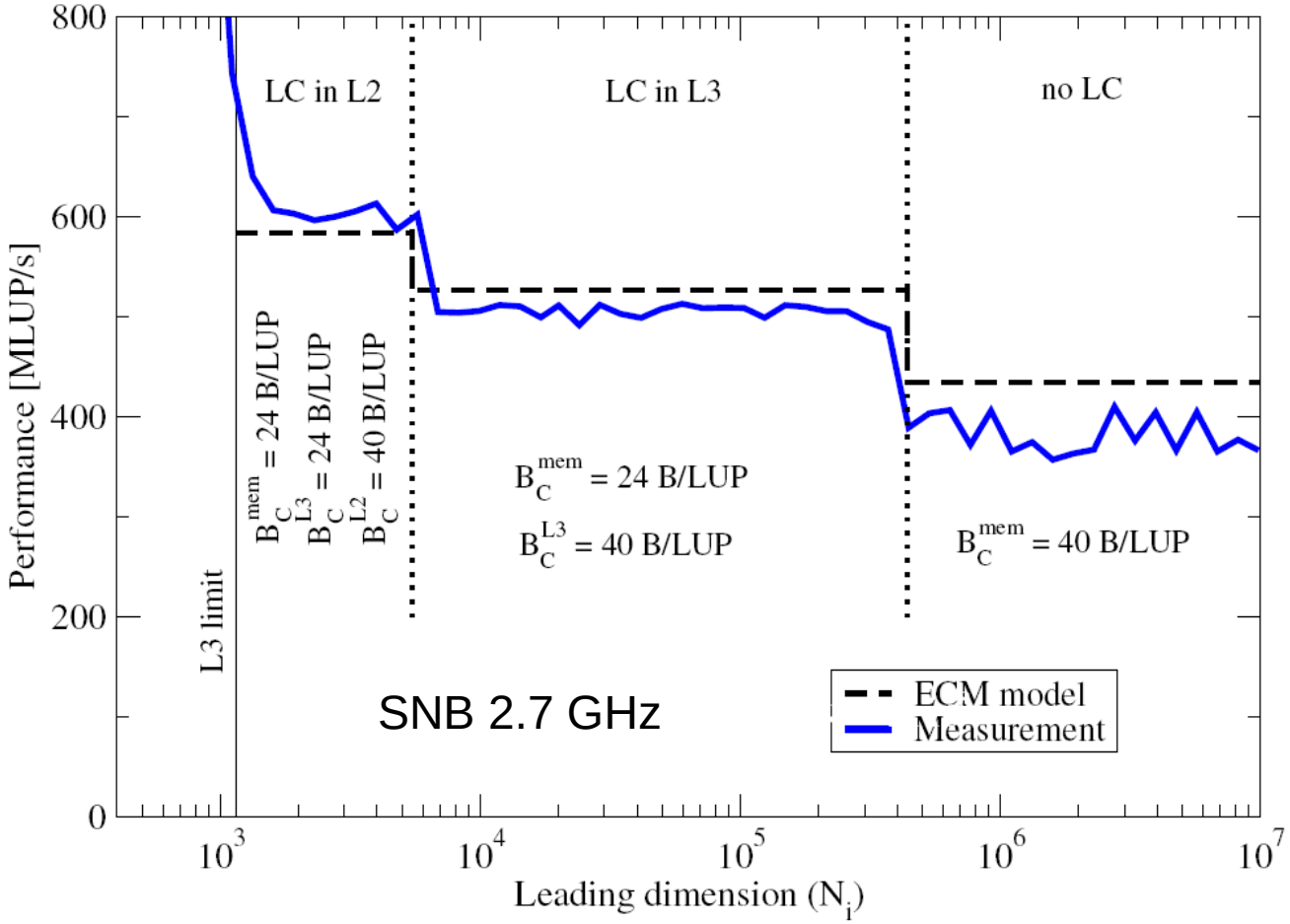
8 iterations (DP):

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



LC = layer condition satisfied in …

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

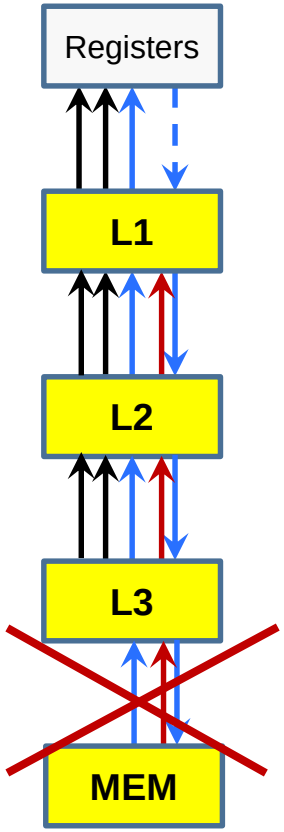


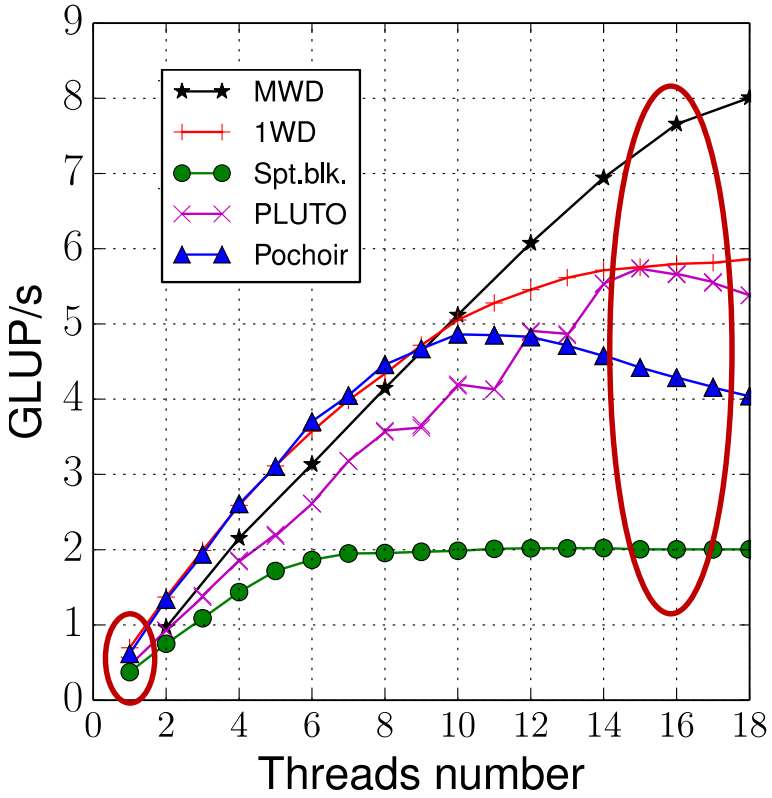
H. Stengel, J. Treibig, G. Hager, and G. Wellein: *Quantifying performance bottlenecks of stencil computations using the Execution-Cache-Memory model*. Proc. ICS15, the 29th International Conference on Supercomputing, June 8-11, 2015, Newport Beach, CA. DOI: 10.1145/2751205.2751240.

Temporal blocking from the resources/modeling point of view

Major step: remove contribution from L3-Mem

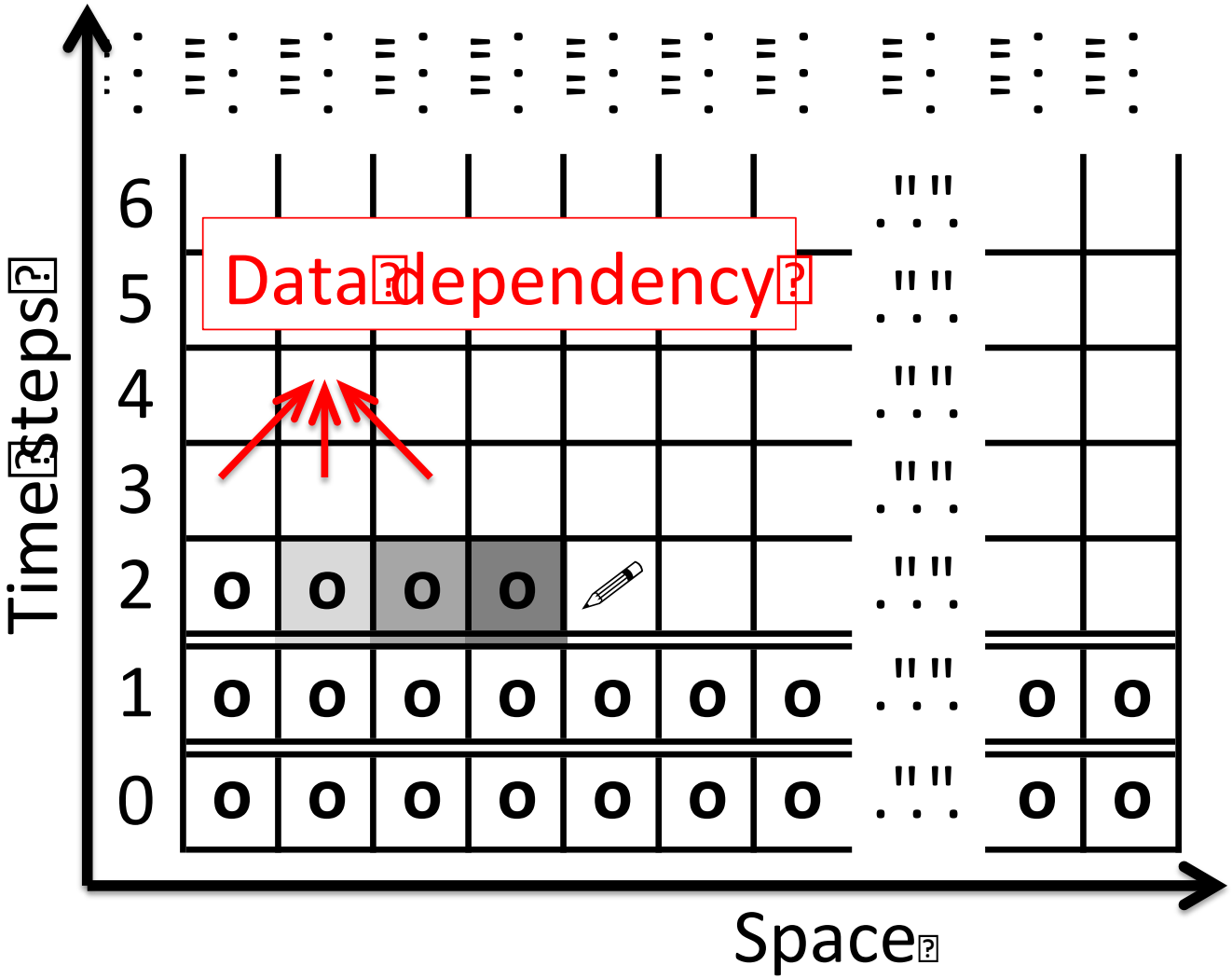
→ Bottleneck is gone
→ Scalable resources
→ Other effects dominate scaling
→ Very limited single-core effect





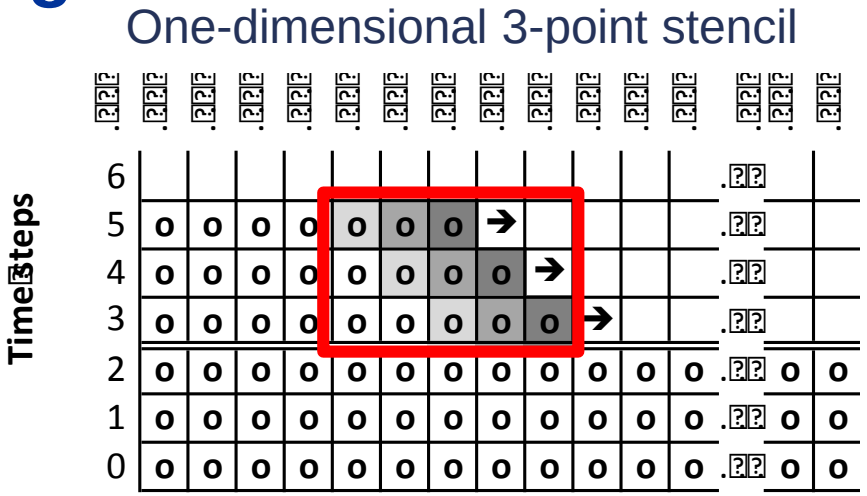
L3 $\{6 \| 8 | 10 | 10 | \cancel{13}\}$ $\{8 \rceil 18 \rceil 28 \rceil \mathbf{28}\}$ cy

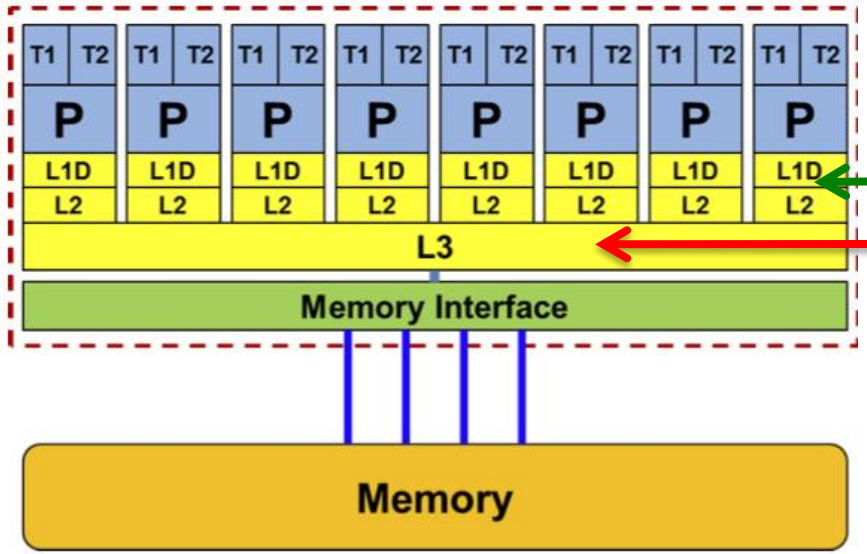
From spatial blocking to temporal blocking: Naïve stencil update order

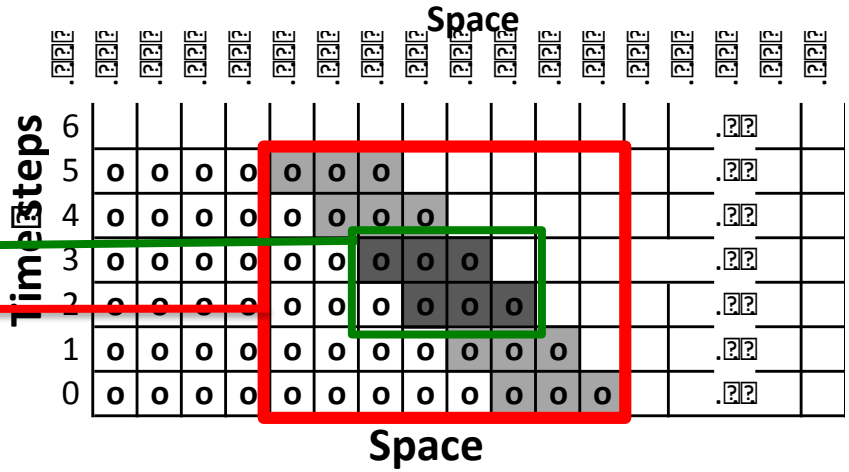


Wavefront temporal blocking

- Data pipelining in the cache:
  - Bring new element to the front
  - Write back result from the back
- Provides maximum data reuse in space-time blocks





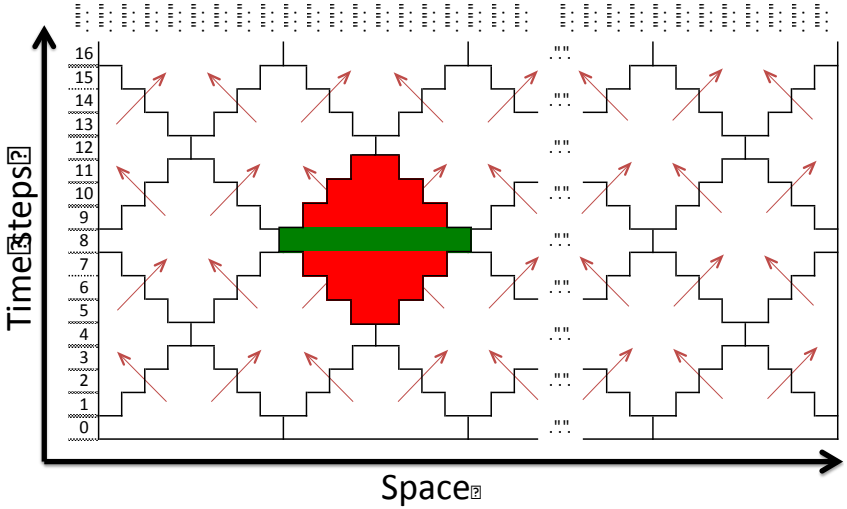


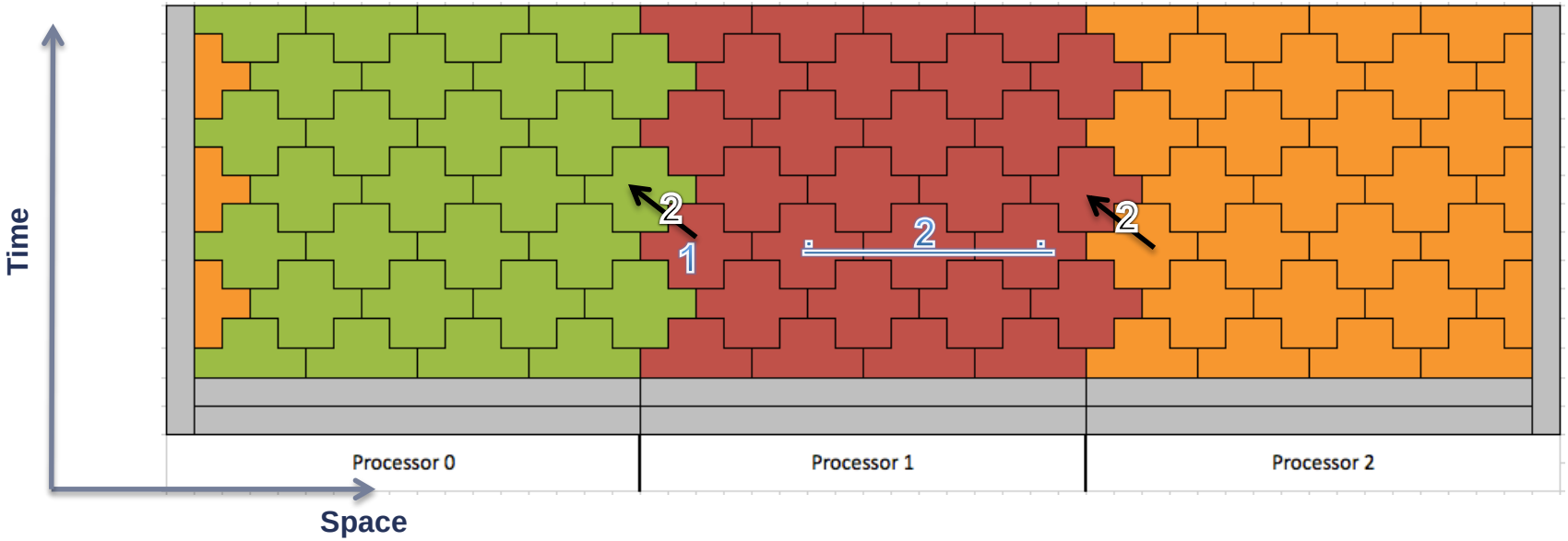
Diamond tiling

Maximum data reuse of loaded block, reducing memory accesses

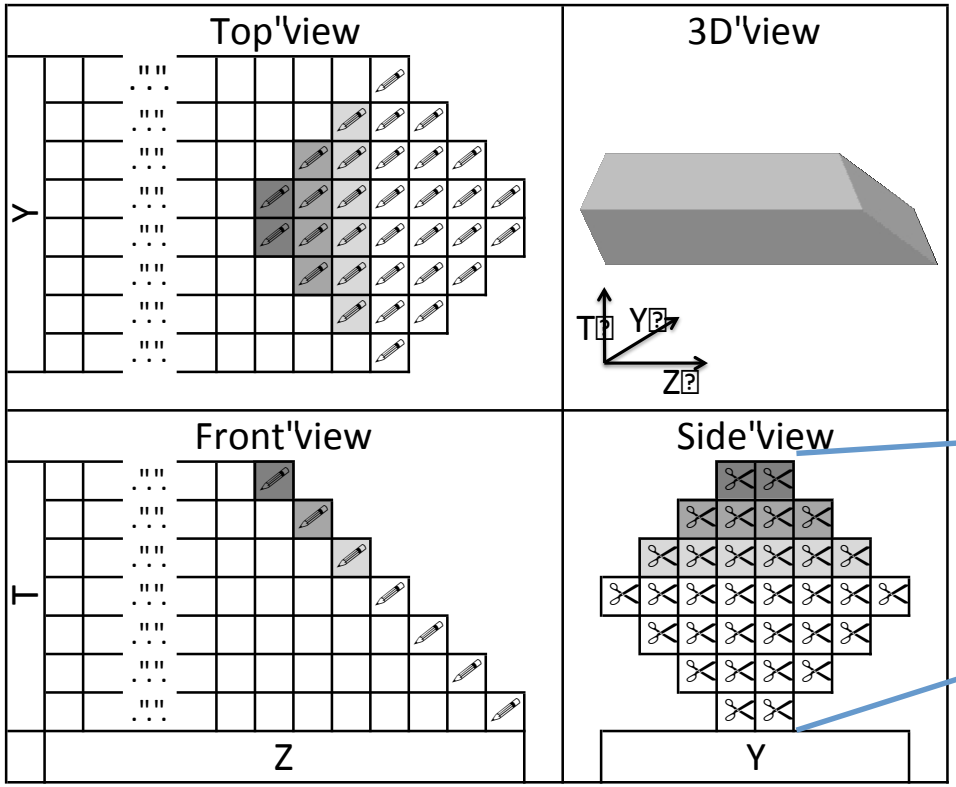
Provides unified tile shape

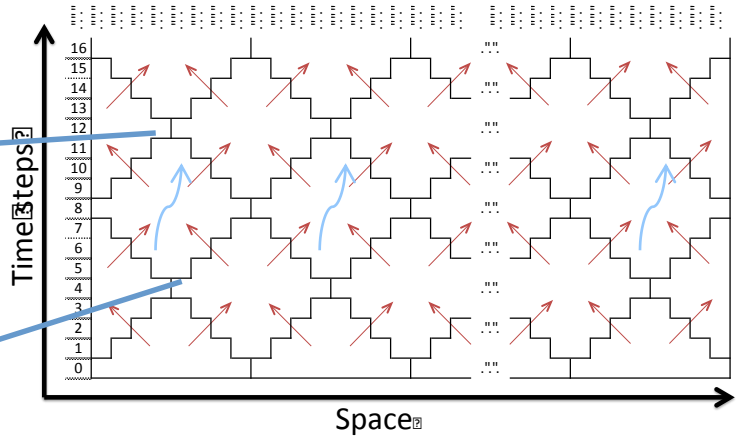
Provides independent space-time blocks to reduce synchronization between threads





Single-core Wavefront temporal blocking + Diamond tiling (1WD)



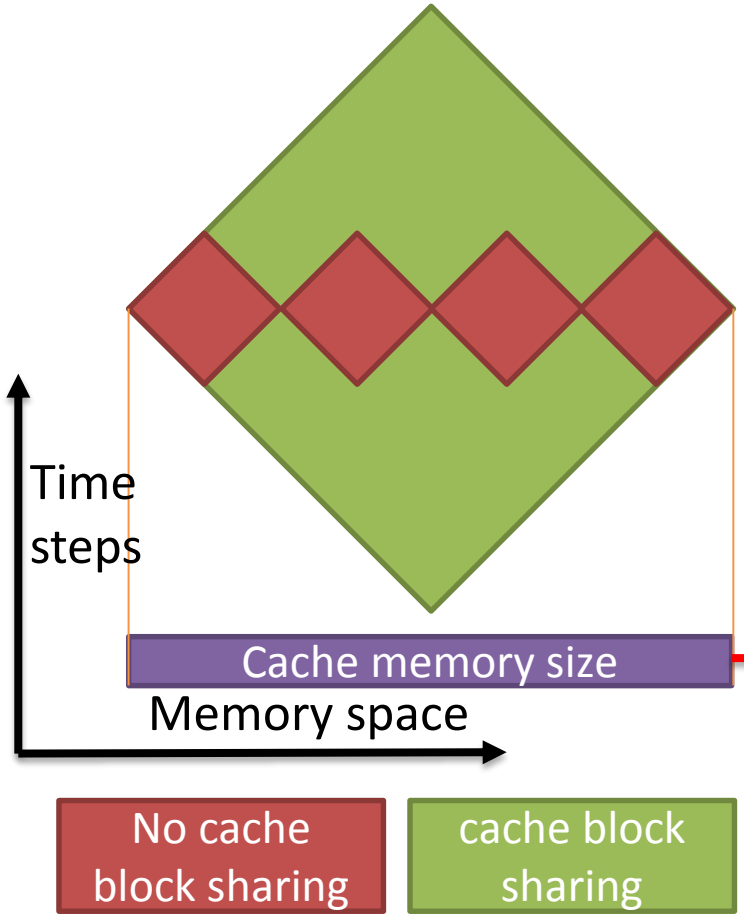


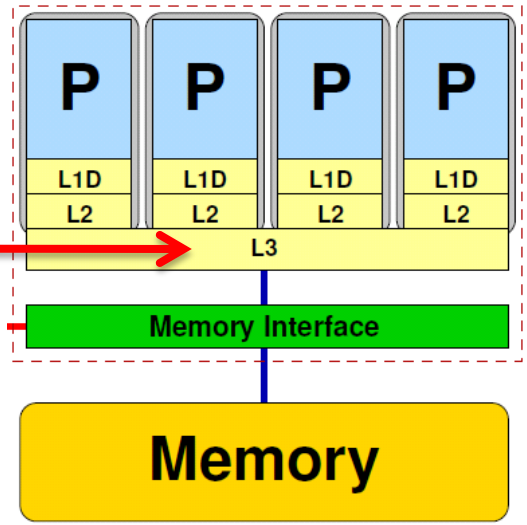
[1] Strzodka *et al.* (2011). *Cache Accurate Time Skewing in Iterative Stencil Computations*. ICPP

[2] Bandishti *et al.* (2012). *Tiling stencil computations to maximize parallelism.* Supercomputing

From 1WD to MWD: Cache block sharing

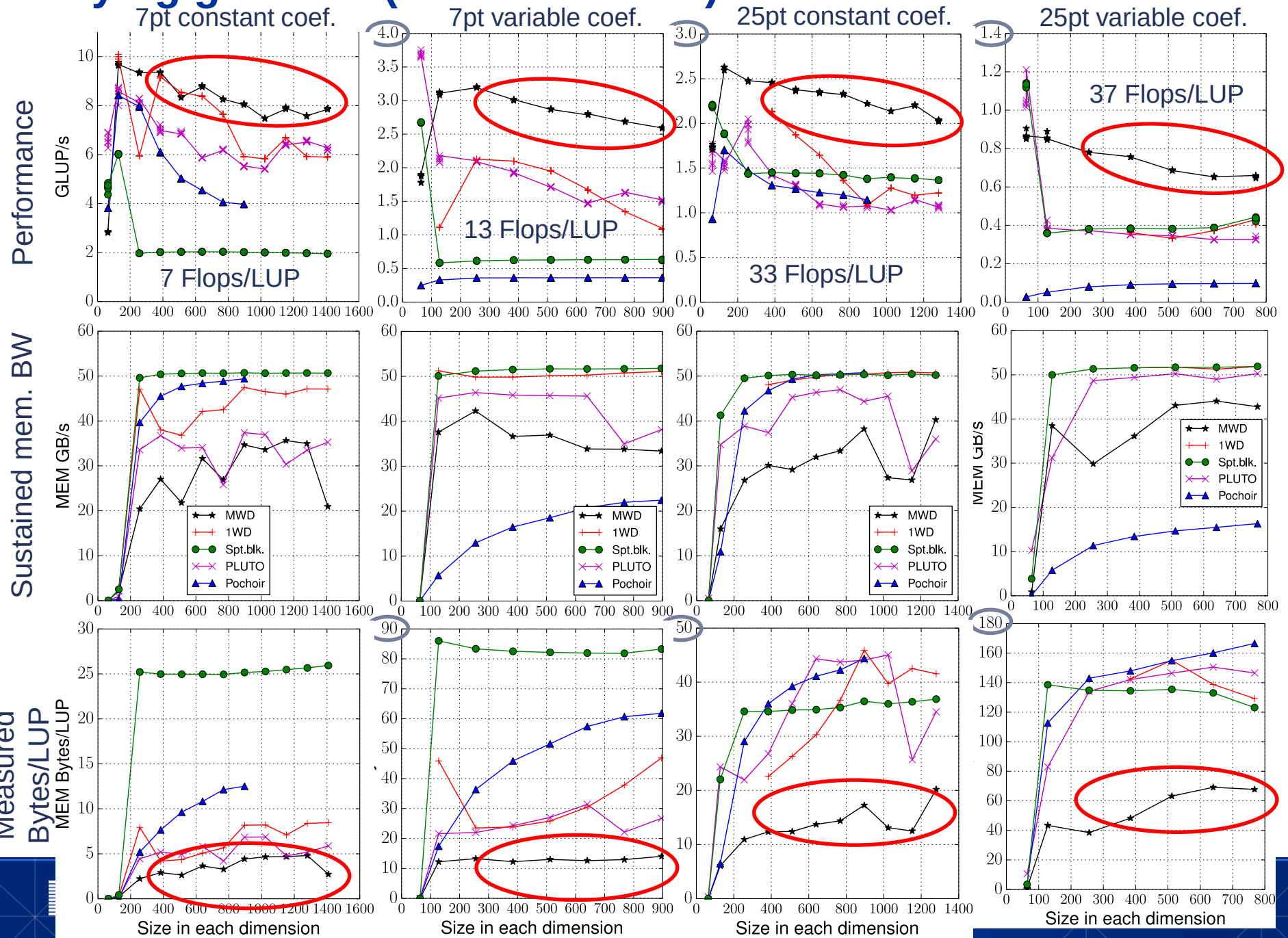
- Reduce the required number of tiles
- Increase the tile size
- More in-cache data reuse
- Less memory bandwidth pressure





G. Wellein, G. Hager, T. Zeiser, M. Wittmann, and H. Fehske: *Efficient temporal blocking for stencil computations by multicore-aware wavefront parallelization.* Proc. COMPSAC 2009, DOI:10.1109/COMPSAC.2009.82

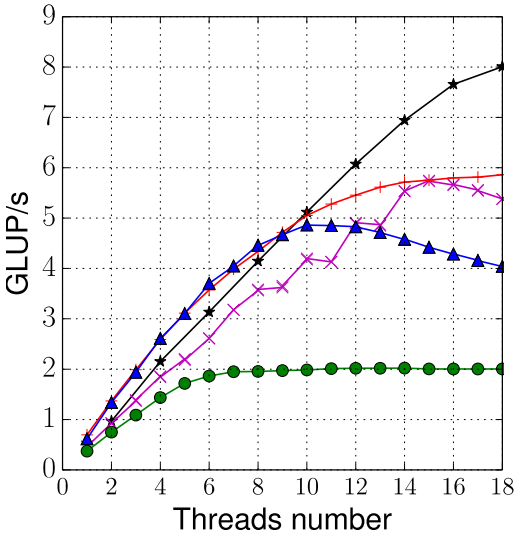
Varying grid size (all frameworks) on 18-core Haswell



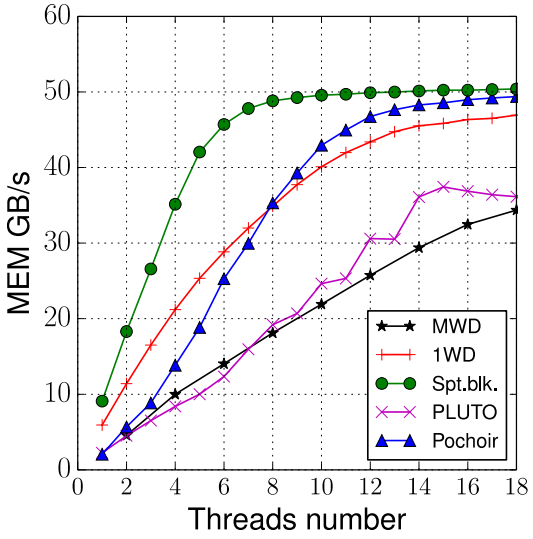
Thread scaling (all frameworks) on 18-core Haswell

7pt constant coefficient stencil

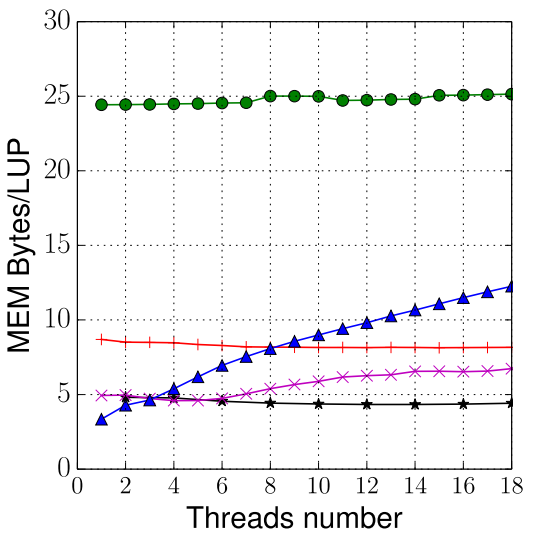
Performance



Sustained mem. BW

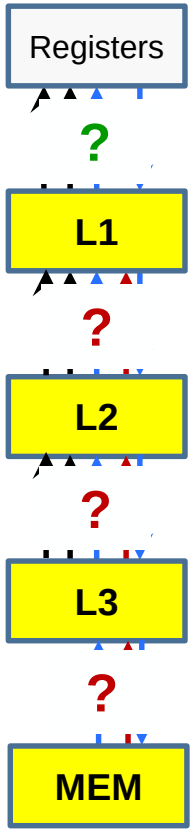


Measured Bytes/LUP



T. M. Malas, G. Hager, H. Ltaief, and D. E. Keyes: *Multi-dimensional intra-tile parallelization for memory-starved stencil computations*. In review. Preprint: arXiv:1510.04995

Phenomenological modeling



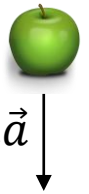
Roadblocks for predictive modeling

- Short(ish) loops → broken steady-state assumption
  → inaccurate in-core prediction
- Finite block sizes → boundary effects
  → inaccurate data traffic prediction

Remedy for traffic prediction: don't predict but measure the traffic

Stencil	Model [cy]	Pred. [GLUP/s]	Meas. [GLUP/s]
7-pt const. coeff.	{12 ‖ 14 | 7 | 7.5 | 1.8}	10	8.0
7-pt var. coeff.	{14 ‖ 21 | 14 | 25 | 4.8}	3.9	2.6
25-pt const. coeff.	{12 ‖ 56 | 20 | 30 | 7.4}	2.5	2.2
25-pt var. coeff.	{12 ‖ 38 | 56 | 50 | 26}	0.71	0.65

What use is phenomenological modeling?



Purely predictive analytic model	Phenomenological analytic model	Curve-fitting "analytic" model
▪ Direct insight into bottlenecks from first principles ▪ Model failures challenge model assumptions or input data ▪ Refinements lead to better insights	▪ Insight into interaction of data transfer & execution ▪ Model failure points to inaccurate in-core model	▪ Yields scalability predictions ▪ Model failure indicates shortcomings of fitting approach ▪ Refinements by using more fit parameters

Further pointers

- (Semi-) Automatic modeling of streaming kernels with Kerncraft
  - https://github.com/RRZE-HPC/kerncraft
- LIKWID toolkit for HPM measurements (and much more)
  - https://github.com/RRZE-HPC/likwid
- Layer condition and block size calculator
  - https://rrze-hpc.github.io/layer-condition/
- Girih test harness for temporally blocked stencil algorithms
  - https://github.com/ecrc/girih

ERLANGEN REGIONAL COMPUTING CENTER

DFG Priority Programme1648

Bavarian Network for HPC

Thank You.

Holger Stengel
Julian Hammer
Jan Eitzinger
Gerhard Wellein

Further references

- J. Treibig and G. Hager: *Introducing a Performance Model for Bandwidth-Limited Loop Kernels*. Proceedings of the Workshop “Memory issues on Multi- and Manycore Platforms” at PPAM 2009, the 8th International Conference on Parallel Processing and Applied Mathematics, Wroclaw, Poland, September 13-16, 2009. Lecture Notes in Computer Science Volume 6067, 2010, pp 615-624. DOI: 10.1007/978-3-642-14390-8_64 (2010).
- J. Treibig, G. Wellein and G. Hager: *Efficient multicore-aware parallelization strategies for iterative stencil computations*. Journal of Computational Science 2, 130-137 (2011). DOI: 10.1016/j.jocs.2011.01.010
- G. Hager, J. Treibig, J. Habich, and G. Wellein: *Exploring performance and power properties of modern multicore chips via simple machine models*. Concurrency and Computation: Practice and Experience, DOI: 10.1002/cpe.3180 (2013).
- M. Wittmann, G. Hager, T. Zeiser, J. Treibig, and G. Wellein: *Chip-level and multi-node analysis of energy-optimized lattice-Boltzmann CFD simulations.* Concurrency and Computation: Practice and Experience 28(7), 2295-2315 (2015). DOI: 10.1002/cpe.3489
- T. M. Malas, J. Hornich, G. Hager, H. Ltaief, C. Pflaum, and D. E. Keyes: *Optimization of an electromagnetics code with multicore wavefront diamond blocking and multi-dimensional intra-tile parallelization*. Proc. IPDPS16, DOI: 10.1109/IPDPS.2016.87

Further references

- M. Wittmann, G. Hager, J. Treibig and G. Wellein: *Leveraging shared caches for parallel temporal blocking of stencil codes on multicore processors and clusters.* Parallel Processing Letters **20** (4), 359-376 (2010). DOI: 10.1142/S0129626410000296
- J. Treibig, G. Hager, H. G. Hofmann, J. Hornegger, and G. Wellein: *Pushing the limits for medical image reconstruction on recent standard multicore processors.* International Journal of High Performance Computing Applications **27**(2), 162-177 (2013). DOI: 10.1177/1094342012442424
- T. M. Malas, G. Hager, H. Ltaief, H. Stengel, G. Wellein, and D. E. Keyes: *Multicore-optimized wavefront diamond blocking for optimizing stencil updates*. SIAM Journal on Scientific Computing **37**(4), C439-C464 (2015). DOI: 10.1137/140991133
- J. Hammer, G. Hager, J. Eitzinger, and G. Wellein: *Automatic Loop Kernel Analysis and Performance Modeling With Kerncraft*. Proc. PMBS15, in conjunction with SC15, November 16, 2015, Austin, TX. DOI: 10.1145/2832087.2832092
- J. Hammer, J. Eitzinger, G. Hager, and G. Wellein: *Kerncraft: A Tool for Analytic Performance Modeling of Loop Kernels*. To appear in Proc. IPTW 2016, the 10th International Parallel Tools Workshop, October 4-5, 2016, Stuttgart, Germany. Preprint: arXiv:1702.04653

Abstract

Many techniques have been devised to improve the performance of stencil algorithms on cache-based multicore CPUs. The main goal is to decouple from the scarce resource of main memory bandwidth, but this is just where the real challenges begin: How can the equally scarce cache space be used most effectively? What is the next bottleneck beyond memory bandwidth? Does it make sense to block for higher-level caches? What is the role of low-level code quality? What is a good parallelization strategy? Some of these questions can be answered by auto-tuning techniques, but others require deeper analysis with the help of analytic performance models. Such models enable us to pinpoint relevant performance issues and sometimes lead to surprising insights. The talk gives an introduction to useful analytic performance models of streaming kernels and how they can by applied to temporally blocked stencil algorithms. Using relevant corner cases we demonstrate how far these models can take us and where they stop being purely predictive. Beyond this point one can still use the principles behind the models to learn more about the bottlenecks or shortcomings of running code by constructing the models not from first principles but from performance counter measurements. We call this approach “phenomenological modeling.” Backed by these concepts we present an analysis and comparison among state-of-the-art stencil frameworks.