Design of Parallel and High-Performance Computing

Fall 2018

Lecture: Roofline model

Instructor: Torsten Hoefler & Markus Püschel

TA: Salvatore Di Girolamo

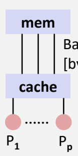
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Roofline model (Williams et al. 2008)

Resources in a processor that bound performance:

- peak performance [flops/cycle]
- memory bandwidth [bytes/cycle]
- <others>

Platform model



Bandwidth β [bytes/cycle] ← carefully measured

- raw bandwidth from manual is unattainable (maybe 60% is)
- Stream benchmark may be conservative

Peak performance π [ops/cycle]

Algorithm model (n is the input size)

Operational intensity I(n) = W(n)/Q(n) =

$$\frac{\text{number of flops (cost)}}{\text{number of bytes transferred between memory and cache}} \text{ [ops/bytes]}$$

Q(n): assumes empty cache;
best measured with performance counters

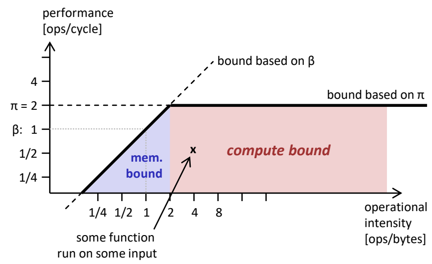
Notes

In general, Q and hence W/Q depend on the cache size m [bytes].
For some functions the optimal achievable W/Q is known:

FFT/sorting: Θ(log(m))
Matrix multiplication: Θ(sqrt(m))

Roofline model

Example: one core with π = 2 and β = 1 and no SSE
ops are double precision flops



Bound based on β?

- assume program as operational intensity of x ops/byte
- it can get only β bytes/cycle
- hence: performance = y ≤ βx
- in log scale: $\log_2(y) \le \log_2(\beta) + \log_2(x)$
- line with slope 1; y = β for x = 1

Variations

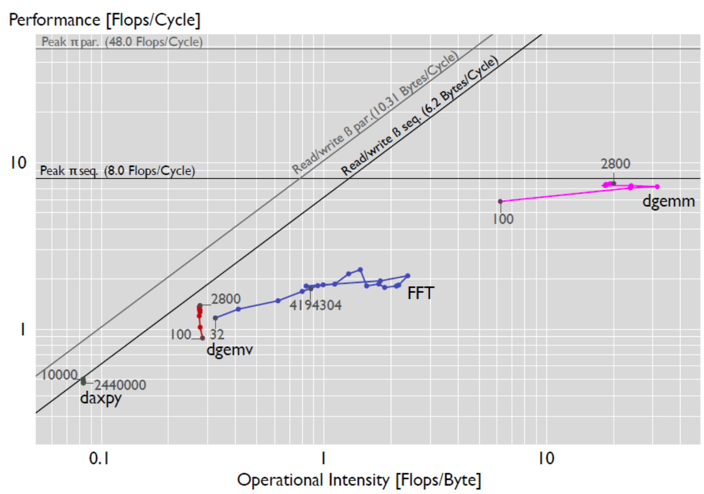
- vector instructions: peak bound goes up (e.g., 4 times for AVX)
- multiple cores: peak bound goes up (p times for p cores)
- program has uneven mix adds/mults: peak bound comes down (note: now this bound is program specific)
- accesses with little spatial locality: operational intensity decreases (because entire cache blocks are loaded)

Roofline Measurements

- **Tool developed in our group**
  (G. Ofenbeck, R. Steinmann, V. Caparros-Cabezas, D. Spampinato)
  http://www.spiral.net/software/roofline.html
- **Example plots follow**
- **Get (non-asymptotic) bounds on I:**
  - daxpy: y = αx+y
  - dgemv: y = Ax + y
  - dgemm: C = AB + C
  - FFT

Roofline Measurements

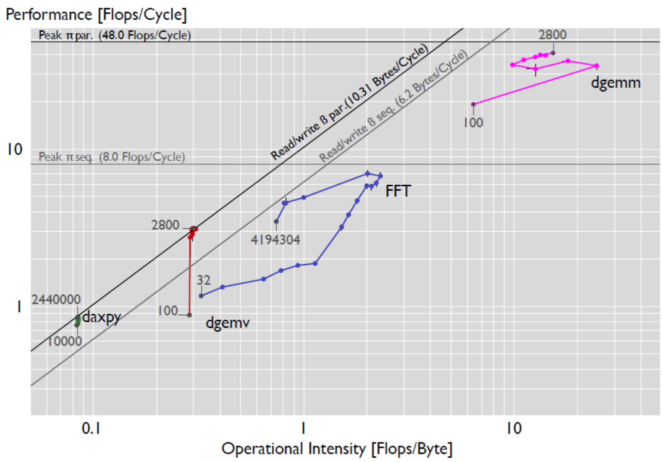
Core i7 Sandy Bridge, 6 cores
Code: Intel MKL, ***sequential***
Cold cache



What happens when we go to parallel code?

Roofline Measurements

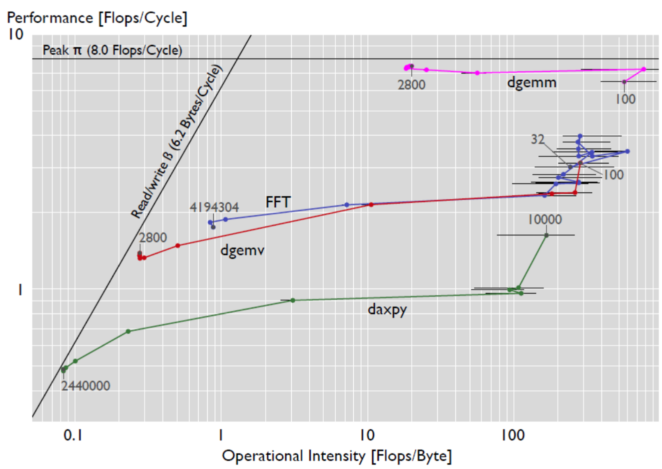
Core i7 Sandy Bridge, 6 cores
Code: Intel MKL, ***parallel***
Cold cache



What happens when we go to warm cache?

Roofline Measurements

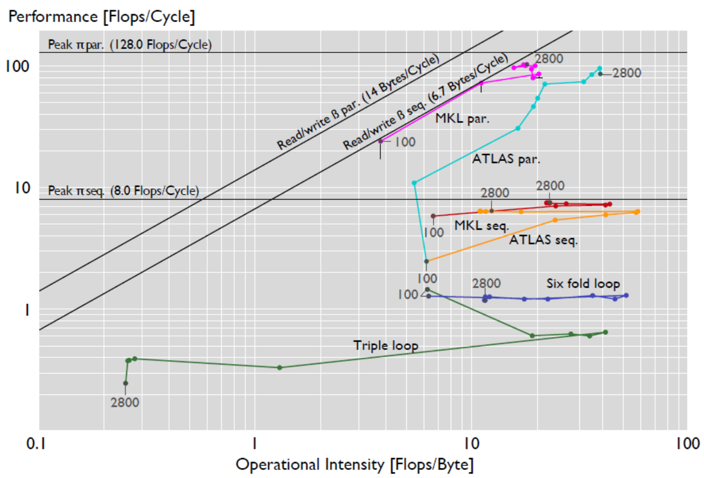
Core i7 Sandy Bridge, 6 cores
Code: Intel MKL, ***sequential***
Warm cache



ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Core i7 Sandy Bridge, 6 cores
Code: Various MMM
Cold cache

Roofline Measurements



MMM: Try to guess the basic shapes

Summary

- **Roofline plots distinguish between memory and compute bound**
- **Can be used on paper**
- **Measurements difficult (performance counters) but doable**
- **Interesting insights:** ***use in your project!***

References

- Samuel Williams, Andrew Waterman, David Patterson
  Roofline: an insightful visual performance model for multicore architectures
  Communications ACM 55(6): 121-130 (2012)
- Georg Ofenbeck, Ruedi Steinmann, Victoria Caparros, Daniele G. Spampinato and Markus Püschel
  Applying the Roofline Model
  Proc. IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), 2014, pp. 76-85
- Victoria Caparros and Markus Püschel
  Extending the Roofline Model: Bottleneck Analysis with Microarchitectural Constraints
  Proc. IEEE International Symposium on Workload Characterization (IISWC), pp. 222-231, 2014