

How to Write Fast Numerical Code

Spring 2019

Lecture: Roofline model

Instructor: Markus Püschel

TA: Tyler Smith, Gagandeep Singh, Alen Stojanov

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Operational Intensity Again

- **Definition:** Given a program P, assume cold (empty) cache

$$\text{Operational intensity: } I(n) = \frac{W(n)}{Q(n)}$$

#flops (input size n) ← $W(n)$
#bytes transferred cache ↔ memory (for input size n) ← $Q(n)$

- **Examples: Determine asymptotic bounds on $I(n)$**
 - Vector sum: $y = x + y$ $O(1)$
 - Matrix-vector product: $y = Ax$ $O(1)$
 - Fast Fourier transform $O(\log(n))$
 - Matrix-matrix product: $C = AB + C$ $O(n)$

Compute/Memory Bound

- A function/piece of code is:
 - *Compute bound* if it has high operational intensity
 - *Memory bound* if it has low operational intensity
- The roofline model makes this more precise
- Blackboard

3

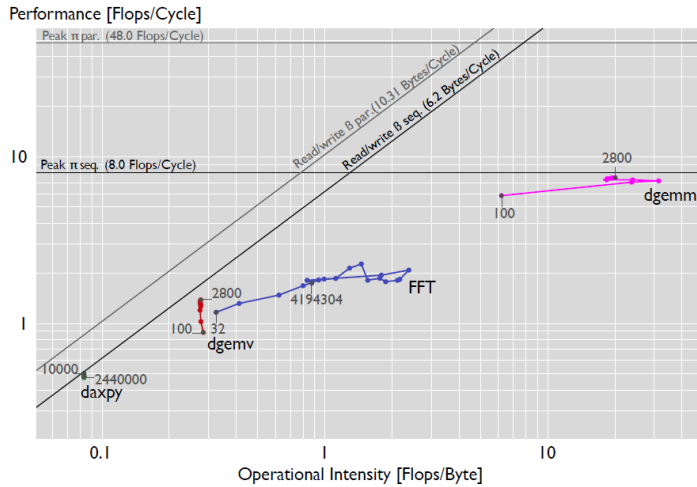
Roofline Measurements

- Tool developed in our group
(G. Ofenbeck, R. Steinmann, V. Caparros-Cabezas, D. Spampinato)
<http://www.spiral.net/software/roofline.html>
- You can use it in your project
- Example plots follow
- Get (non-asymptotic) bounds on I:
 - daxpy: $y = \alpha x + y$
 - dgemv: $y = Ax + y$
 - dgemm: $C = AB + C$
 - FFT

4

Roofline Measurements

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

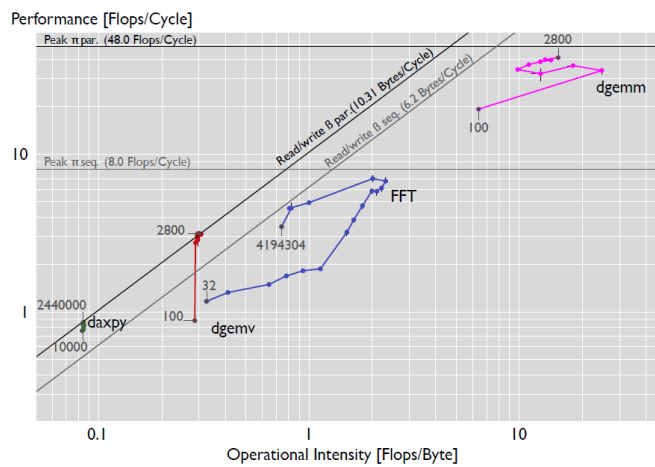


What happens when we go to parallel code?

5

Roofline Measurements

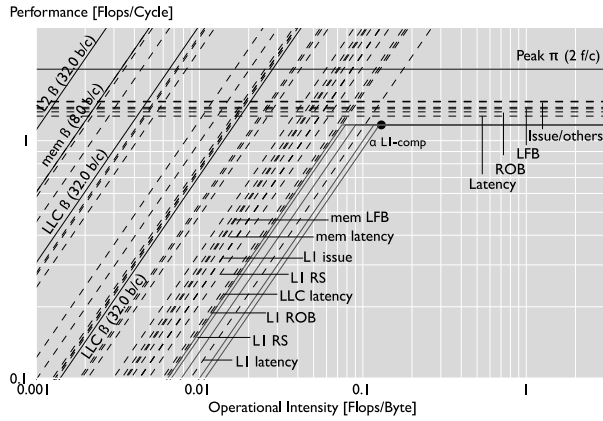
Core i7 Sandy Bridge, 6 cores
Code: Intel MKL, *parallel*
Cold cache



What happens when we go to warm cache?

6

Generalized Roofline Model



Website and tool: <https://acl.inf.ethz.ch/research/ERM/>

Caparrós Cabezas and P. **Extending the Roofline Model: Bottleneck Analysis with Microarchitectural Constraints**, Proc. IISWC, pp. 222-231, 2014

9

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!*

10