Operational Intensity Again

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

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

- #flops (input size \( n \))
- #bytes transferred cache ↔ memory (for input size \( n \))

Asymptotic bounds on \( I(n) \)

- Vector sum: \( y = x + y \) \( \mathcal{O}(1) \)
- Matrix-vector product: \( y = Ax \) \( \mathcal{O}(1) \)
- Fast Fourier transform \( \mathcal{O}(\log(n)) \)
- Matrix-matrix product: \( C = AB + C \) \( \mathcal{O}(n) \)

Cache lecture
\( y = \text{size LLC (last level cache)} \)
Known to be optimal

\( \gamma = \text{size LLC (last level cache)} \)
Known to be optimal
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

Roofline model/plot *(Williams et al. 2008)*

<table>
<thead>
<tr>
<th>Platform model</th>
<th>Algorithm/program model (n is the input size)</th>
</tr>
</thead>
<tbody>
<tr>
<td>mem → cache → mem</td>
<td>mem → cache → mem</td>
</tr>
<tr>
<td>Bandwidth $\beta$ ← carefully measured (bytes/cycle)</td>
<td>Data movement: $Q(n)$ [bytes]</td>
</tr>
<tr>
<td>raw bandwidth from manual unattainable (maybe 60% is)</td>
<td>Runtime: $T(n)$ [cycles]</td>
</tr>
<tr>
<td>$\beta$ ≥ Q/T = (W/T)/(W/Q) = P/I</td>
<td></td>
</tr>
<tr>
<td>p cores</td>
<td>Derived:</td>
</tr>
<tr>
<td>Each with peak performance $\pi$ [flops/cycle]</td>
<td>Operational intensity $I(n) = W(n)/Q(n)$ [flops/byte]</td>
</tr>
<tr>
<td></td>
<td>Performance: $P(n) = W(n)/T(n)$ [flops/cycle]</td>
</tr>
</tbody>
</table>

Example: one core, $\pi = 2$, $\beta = 1$, no SIMD

Bound based on $\beta$:

$\beta ≥ Q/T = (W/T)/(W/Q) = P/I$

In log scale: $\log_2(P) ≤ \log_2(\beta) + \log_2(I)$

- Bound based on $\pi$: $P ≤ \pi$
- Bound based on $\beta$: $P ≤ \beta I$
- Compute bound
- Memory bound
- Some program run on some input

<table>
<thead>
<tr>
<th>$P(n)$ [flops/cycle]</th>
<th>$I(n)$ [flops/byte]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/4 1/2 1</td>
<td>4 2 1</td>
</tr>
<tr>
<td>$\pi/\beta$</td>
<td>$\pi$</td>
</tr>
</tbody>
</table>
Roofline Plots

- What happens if we introduce 4-way SIMD?

If $\beta$ does not change:
more programs become memory bound

Roofline Plots

- What if a program has an uneven mix of operations (e.g., 20% mults and 80% adds)?

A tighter roof may hold for this program
(depends on units and ports)
Roofline Measurements

- Tool developed in our group (bot may need an update)
  (G. Ofenbeck, R. Steinmann, V. Caparros-Cabezas, D. Spampinato)
  http://www.spiral.net/software/roofline.html
- Example plots follow
- Estimate operational intensity $I = \frac{W}{Q}$ (cold cache):
  - daxpy: $y = ax + y$  $W = 2n$  $Q = 3n$ doubles = $24n$ bytes  $I = \frac{1}{12}$
  - dgemv: $y = Ax + y$  $W = 2n^2$  $Q = n^2$ doubles = $8n^2$ bytes  $I = \frac{1}{4}$
  - dgemm: $C = AB + C$  $W = 2n^3$  $Q \geq 4n^2$ doubles = $32n^2$ bytes  $I \leq n/16$
  - FFT

Note:
- For daxpy and dgemv, $Q$ is determined by compulsory misses.
- For dgemm, more misses than compulsory misses occur for larger sizes.
  If $3n^2 \leq \gamma$ (cache size), equality should hold above.

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

What happens when we go to parallel code?
Roofline Measurements

What happens when we measure with warm cache?

Roofline Measurements

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

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

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

Generalized Roofline Model

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

Summary

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