Design of Parallel & High Performance Computing

Reasoning about Performance II

Roofline Model - Balance Principles

5. Roofline Model (Williams et al. 2008)

resources in a microarchitecture that bound performance:

- peak performance π [flops/cycle]
- memory bandwidth β [bytes/cycle]

associated program features

- work W [flops]
- data transferred cache $\leftrightarrow$ mem Q [bytes]

for a given problem: $\min W$ = work/time complexity
$\min Q$ = I/O complexity

Operational intensity I: Given a program, assume empty cache:

$$I(n) = W(n)/Q(n)$$

Intuition: high I $\leftrightarrow$ compute bound
low I $\leftrightarrow$ memory bound

Example: asymptotic bounds on I

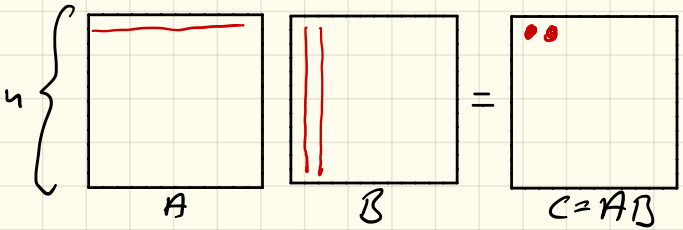
		$I(n)$
vector sum	$y = x + y$	$O(1)$
matrix-vector product	$y = Ax$	$O(1)$
fast Fourier transform		$O(\log n)$
matrix-matrix product	$C = AB$	$O(n)$

Operational Intensity: Example Matrix Multiplication

Assumptions: cache size $\gamma \ll n$, cache block = 8 doubles, 1 cache

We want to estimate Q. $W = 2n^3$ flops

1.) Triple loop:



1. entry of C: $n+8n$ doubles

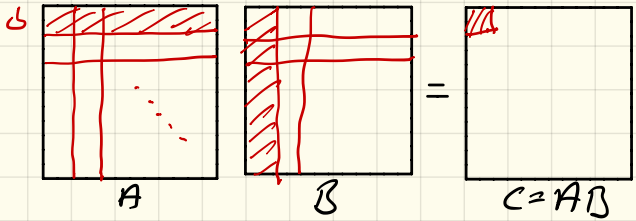
2. entry of C: same

. . . . .

total: $9n^3$ doubles

$I(n) = O(1)$

2.) Blocked: $s|b$, $3s^2 \le \gamma$



1. block of C: $2ns$ doubles

2. block of C: same

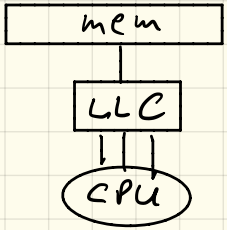
. . .

total: $2ns \cdot \left(\frac{n}{b}\right)^2 = \frac{2n^3}{b}$ doubles

$b = \sqrt{\frac{\gamma}{3}} \Rightarrow I(n) = \Theta(\sqrt{\gamma})$

Roofline Model (Williams et al. 2008)

Computer:



bandwidth β [bytes/cycle]

peak perf. π [flops/cycle]

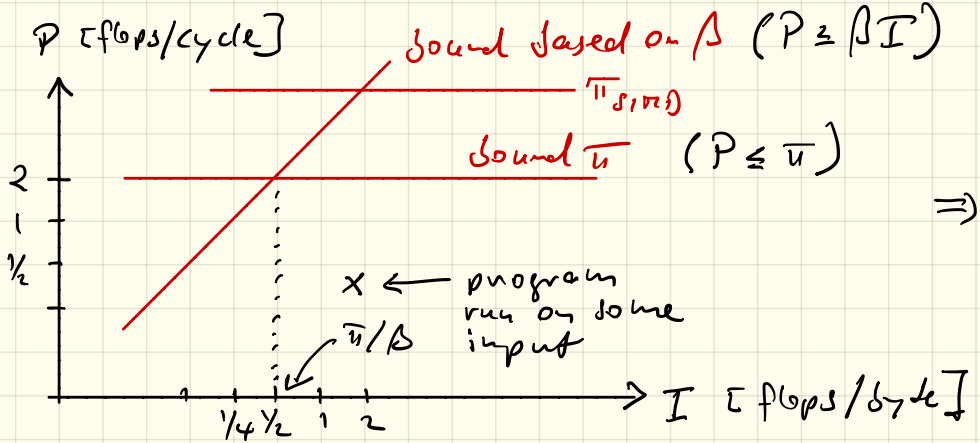
Program:

$I = W/Q$ [flops/byte]

T = runtime [cycles]

$P = W/T$ (performance) [flops/cycle]

Roofline plot: (example $\pi = 2$, $\beta = 4$)



reason: $\frac{P}{I} = \frac{W/T}{W/Q} = \frac{Q}{T} \leq \beta$

$P \leq \beta \cdot I$

$\Rightarrow \log P \leq \log I + \log \beta$

$I = 1 \Rightarrow P \leq \beta$

Operational Intensity: Upper bounds

$I(n) = W(n) / Q(n)$ — #flops / #bytes mem ↔ cache

x, y: length n, A, B, C: $n \times n$, α: scalar

		asymptotic	best	exact
daxpy	$y = \alpha x + y$	$O(1)$	$O(1)$	$\leq 1/12$
dgemv	$y = Ax + y$	$O(1)$	$O(1)$	$\leq 1/4$
fft		$O(\log n)$	$\Theta(\log Z)$	——
dgemm	$C = AB + C$	$O(n)$	$\Theta(\sqrt{Z})$	$\leq \frac{n+1}{16}$

Z = cache size

should be sharp if $3n^2 \cdot 8 \leq Z$

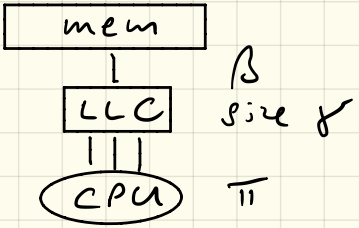
Example: $Z = 128\,\text{kB}$

$\Rightarrow n \leq 200$

Cache to slides

6. Balance Principles I (Kung 86)

Computer:



Program:

Q_γ: data transfer mem ↔ LLC

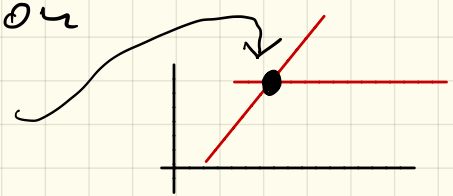
W: work = # ops

The computer is called **balanced** if

compute time = data transfer time

i.e., assuming perfect utilization

$$\frac{W}{\pi} = \frac{Q_\gamma}{\beta} \iff \frac{\pi}{\beta} = \frac{W}{Q_\gamma}$$



assume $\pi \to a\pi$ increases; how to rebalance?

a.) $\beta \to a \cdot \beta$ ✓ but β grows slower than π!

b.) $\gamma \to x \cdot \gamma$ what is x?

Example 1: Matrix multiplication $C = AB + C$

algorithm with optimal $I = \frac{W}{Q_\gamma} = \Theta(\sqrt{\gamma})$

$$\frac{\pi}{\beta} = \frac{W}{Q_\gamma} = \Theta(\sqrt{\gamma}) \qquad \pi \to a\pi, \ \gamma \to a^2 \cdot \gamma$$

Example 2: FFT / Sorting

algorithm with optimal $I = \Theta(\log \gamma)$

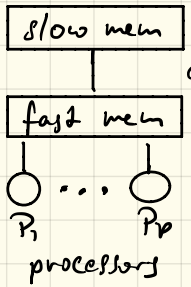
$$\frac{\pi}{\beta} = \frac{W}{Q_\gamma} = \Theta(\log \gamma) \qquad \pi \to a\pi, \ \gamma \to \gamma^a$$

Both are unrealistic.

7. Balance principles II (Czechowski et al. 2011)

Goal: more detailed principles for multicores, algorithm/architecture codesign, assessment of HW trends

Computer:



α: latency, β: throughput, size γ, transaction size λ

each peak perf π

Algorithm:

PRAM: W = work, D: depth

$Q_{\gamma,\lambda}$: mem. transfers of size λ

assumptions: optimal W, W/Q

How to get $Q_{p,\gamma,\lambda}$ from $Q_{\gamma,\lambda}$?

- there are general bounds and some direct results

Processor is balanced if

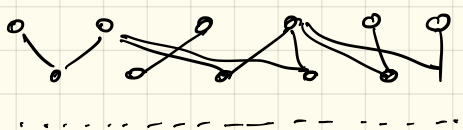
$T_{mem} \leq T_{comp}$ ("compute-bound")

Hong:

$$\frac{Q \cdot \lambda}{\beta} \leq \frac{W}{p \cdot \pi} \iff \frac{p\pi}{\beta} \leq \frac{W}{Q\lambda}$$

Derivation principles:

1.) Estimate T_{mem}: Idea divide DAG in levels



$\Sigma = Q_{\gamma,\lambda}$

In each level α-β model: $\alpha + \frac{q_i \cdot \lambda}{\beta}$

$$\Rightarrow T_{mem} \approx \sum_{i=1}^{D} \left(\alpha + \frac{q_i \cdot \lambda}{\beta}\right) = \alpha \cdot D + \frac{Q}{\beta} \qquad Q = Q_{p,\gamma,\lambda}$$

2.) Estimating T_{comp}: Brent's theorem

$$T_{comp} \approx \left(D + \frac{W}{p}\right) \cdot \frac{1}{\pi}$$

Balance: $T_{mem} \leq T_{comp}$

mem. par. computer

parallelism computer

$$\Leftrightarrow \frac{p\pi}{\beta}\left(1 + \frac{\alpha\beta/\lambda}{Q/D}\right) \leq \frac{W}{Q \cdot \lambda}\left(1 + \frac{p}{W/D}\right)$$

Kung

mem. par. algorithm

parallelism algorithm

Example 1: matrix multiplication

use $Q \geq \frac{W}{\sqrt{\Sigma \cdot \lambda \sqrt{\gamma p}}}$ (Irony et al. 2004) $\Rightarrow I(n) = O\left(\sqrt{\frac{\gamma}{p}}\right)$

$\Rightarrow$ balance principle $\frac{p\pi}{\beta} \in O\left(\sqrt{\frac{\gamma}{p}}\right)$

$\pi \to a \cdot \pi$ rebalance: $\beta \to a\beta$ or $\gamma \to a^2\gamma$

$p \to a \cdot p$ " : $(\beta \to a\beta$ and $\gamma \to a\gamma)$ or $(\gamma \to a^3\gamma)$

Example 2: sorting/FFT

$\Rightarrow \frac{p\pi}{\beta} \in O\left(\log \frac{\gamma}{p}\right)$

Back to strokes