I/O complexity

Goal: Analyze optimality of algorithms w.r.t. data movement

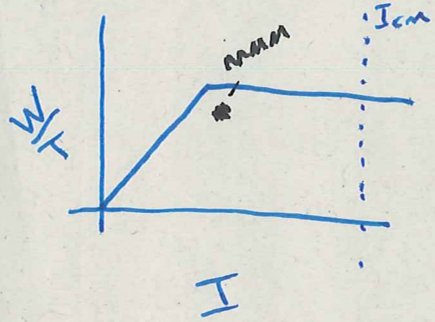
W - work, T - time

Q - reads + writes

$I = \frac{W}{Q}$ (intensity)

γ - size of cache

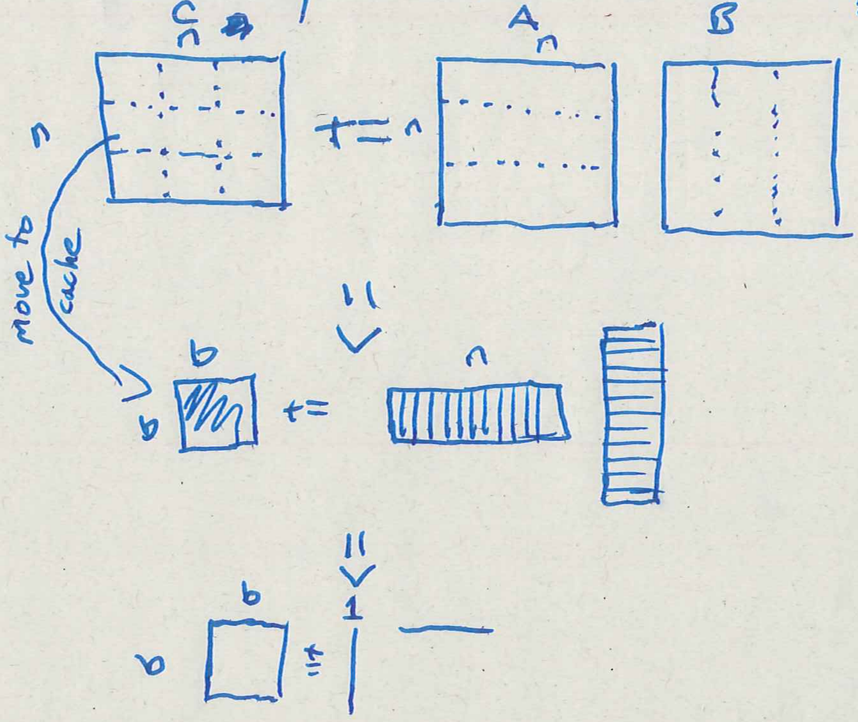
Roofline model



$Q \geq Q_{cm} \Rightarrow I \geq I_{cm}$

Ordinary MMM

A, B, C - n×n matrices



Per block of C:

read b×n panel of A, n×b panel of B

$\frac{n}{b} \cdot \frac{n}{b}$ blocks of C

I/O cost:

$$\underbrace{\frac{2n^3}{b}}_{\text{read A + read B}} + \underbrace{2n^2}_{\text{read/write C}}$$

I/O cm:

$4n^2$

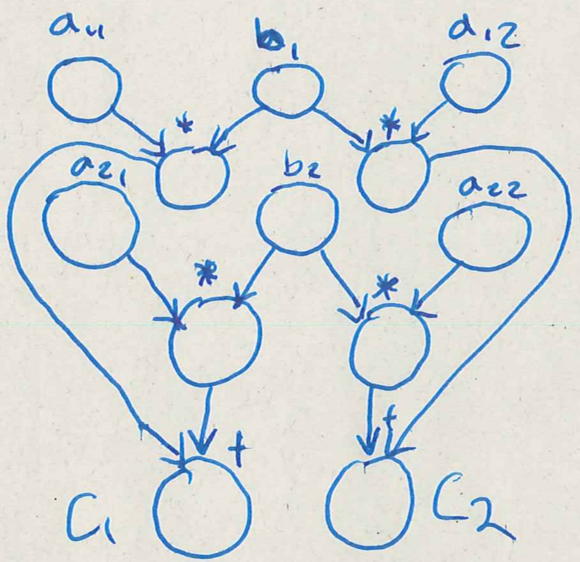
CDAG Model

Vertex - atom of data

edge - dependency

MVM:

$$\begin{bmatrix} c_1 \\ c_2 \end{bmatrix} := \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$



Red - blue pebble game (Hong & Kung 1981)

Start - blue pebbles on inputs

R_1 - Input. place a red pebble on a vertex w/ a blue pebble

R_2 - Output. place a blue pebble on a vertex w/ a red pebble

R_3 - Computation place a red pebble on a vertex whose immediate predecessors have red pebbles

R_4 - Delete any pebble

End: blue pebbles on outputs

Game Moves:

Segment 0 : Segment 1

$R_1(a_{11})\ R_1(b_1)\ R_3(*_{11})\ R_1(a_{12})\ R_4(a_{11})\ R_3(*_{12})\ R_4(b_1)\ R_4(a_{12})\ R_1(a_{12})\ \vdots\ R_1(b)\ R_3(*_{21})\ldots$

- S-span theorem (simplified)
  - Partition game into segments
  - Segment 0 starts at beginning
  - γ R_1+R_2 in segment => end current segment, begin next.
  - Suppose h such segments.
  - index by $i \in \{0, \dots, h-1\}$

I/O in segment i:

$$Q_i = \begin{cases} \gamma, & i < h-1 \\ ?, & i = h-1 \end{cases} \implies Q \geq \gamma(h-1)$$

Work in segment i:

$$W_i \leq W_{max}$$

↑ upper bound on any possible w_i $\implies h \geq \frac{W}{W_{max}}$

$$Q \geq \gamma\left(\frac{W}{W_{max}} - 1\right)$$

MMM lower bound

$W = n^3$ multiplications.

$$Q_{MMM} \geq \gamma\left(\frac{n^3}{W_{max}} - 1\right)$$

What is W_{max}?

N_A elem. A,

N_B elem B.

Output N_c elem. C

Beginning of segment:

$\leq \gamma$ vertices w/ red pebbles

During segment:

$\leq \gamma$ vertices placed by red pebbles.

$$N_A \leq 2\gamma, \quad N_B \leq 2\gamma$$

End of Segment:

$\leq \gamma$ elem w/ red pebbles

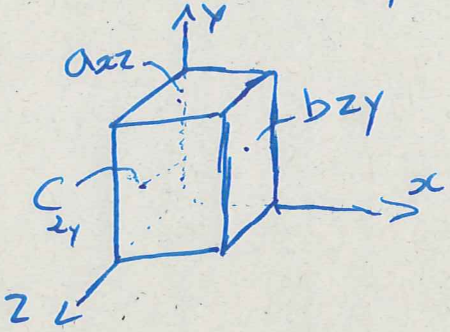
$\leq \gamma$ rule R2

$$N_C \leq 2\gamma$$

Geometric problem:

Multiplications as points in 3D space

$$c_{xy} := a_{xz}\, b_{zy}$$



Project set of points in x dimension:

elem. of b

y dimension: elem. a

z " " c

3D Loomis-Whitney Inequality

Let V be a finite set w/ elem in $\mathbb{Z}^3$

let V_x, V_y, V_z be orthogonal projections of V onto the coordinate planes.

$$|V| \leq \sqrt{|V_x| \cdot |V_y| \cdot |V_z|}$$

Applying Loomis-Whitney Inequality (Irony et. al 2004)

$$W_{max} \le \sqrt{N_A \cdot N_B \cdot N_C} \le \sqrt{8\gamma^3}$$

substitute

$N_A, N_B, N_C \le 2\gamma$

$$Q_{MMM} \ge \gamma\left(\frac{n^3}{\gamma\sqrt{8\gamma}} - 1\right) = \frac{n^3}{\sqrt{8}\sqrt{\gamma}} - \gamma$$

Can we improve the lower bound?

Observation 1: We can change the problem

Assume all computation is performed with FMAs

(claim: any game played on an MMM CDAG can be transformed into one that multiplies elements of a, b, and adds them to an element of c immediately)

FMA: $c_{ij} \pm a_{ip} \cdot b_{pj}$

all inputs

$$N_A + N_B + N_C \le 2\gamma$$

Find constrained global maximum of $\sqrt{N_A N_B N_C}$

$$\Rightarrow N_A = N_B = N_C = \frac{2\gamma}{3} \Rightarrow Q_{MMM} \ge \frac{3\sqrt{3}}{2\sqrt{2}} \frac{n^3}{\sqrt{\gamma}} - \gamma$$

Observation 2:

Number of $R_1 + R_2$ during a ~~phase~~ segment is arbitrary

let x = # of R_1s + R_2s in a segment

- free variable

$$N_A + N_B + N_C \le \gamma + x$$

$$W_{max} \le \frac{(\gamma + x)\sqrt{\gamma + x}}{3\sqrt{3}}$$

$$Q \ge x\left(\frac{W}{W_{max}} - 1\right)$$

let $x = 2\gamma$.

$$W_{max} = \gamma\sqrt{\gamma}$$

$$Q_{MMM} \ge \frac{2n^3}{\sqrt{\gamma}} - 2\gamma$$

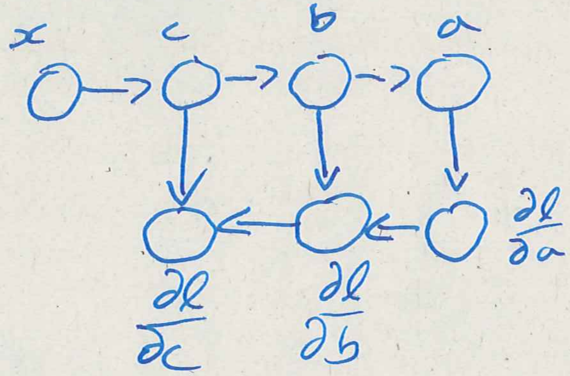
Recomputation Example - Neural Networks

Model:

$f(x) = a \circ b \circ c(x)$

$a, b, c : \mathbb{R}^n \to \mathbb{R}^n$

W_a - weights of a
W_b - " " b
W_c - " " c



Goal: fit model to data. Minimize loss function $\ell(f)$

Training: Stochastic gradient descent (SGD)

- pick random x
- calculate gradient of ℓ at x w.r.t. model parameters e.g $\frac{\partial \ell}{\partial W_a}$
- modify parameters based on gradient

- Chain rule

$$\frac{\partial \ell}{\partial w_a} = \underset{1\times n}{\frac{\partial \ell}{\partial a}} \cdot \underset{n \times m}{\frac{\partial a}{\partial W_a}}$$

$$\frac{\partial \ell}{\partial w_b} = \left(\underset{1\times n}{\frac{\partial \ell}{\partial a}} \cdot \underset{n \times n}{\frac{\partial a}{\partial b}}\right) \cdot \underset{n \times m}{\frac{\partial b}{\partial W_b}}$$

$$\frac{\partial \ell}{\partial w_c} = \left(\left(\frac{\partial \ell}{\partial a} \cdot \frac{\partial a}{\partial b}\right) \cdot \frac{\partial b}{\partial c}\right) \cdot \frac{\partial c}{\partial w_c}$$

Group from left to right
(MVM vs. MMM)

Tradeoffs:

Method 1: Complete Memoization
(assume h stages)

- memory $O(h)$
- time $O(h)$

Method 2: complete recomputation

- memory $O(1)$
- time $O(h^2)$

Method 3 - partial memoization/recomputation

- recompute every k stages

- memory $O(h/k)$
- time $O(h \cdot k)$