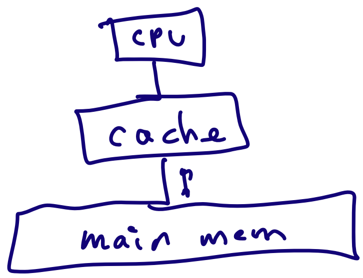
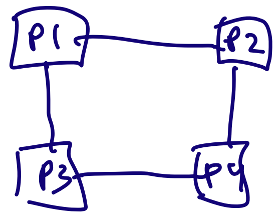
Welcome to Ma 221! Lecture 9, Sep 13

Goal: Understand real cost of algs.

Moving data, not flops, most expensive





Simple Model of Comm. Costs

Bandwidth (bw), Latency

Intuition: freeway from Berkeley to Sacramento

BW = # cars/hour that can go from B → S

#cars/hour = density (# cars/mile/lane)
× velocity (miles/hr)
× lanes

Latency = how long it takes 1 car to get from B → S

time (hrs) = distance (miles) / velocity $\left(\frac{\text{miles}}{\text{hour}}\right)$

So minimum time to move n cars from B → S assuming they all travel in one "convoy" close together

time (hrs) = time for 1^{st} car
+ time for remaining cars
= latency + n/BW

Same formula for moving data:
w words from DRAM to cache
= latency + w/BW
assuming all words in one "message"

Moving w words in m messages costs
m· latency + w/BW

Notation $m \cdot \alpha + w \cdot \beta$ = comm cost
α = latency
β = 1/BW
γ = time per flop
f = # flops

Total time: $f \cdot \gamma + m \cdot \alpha + w \cdot \beta$
Today $\gamma \ll \beta \ll \alpha$ (picture!)

Flops dominate: $f \cdot \gamma \geq m\alpha + w \cdot \beta$

Comm dominates $f\gamma < m \cdot \alpha + w \cdot \beta$

Notation: Computational Intensity
$= q = \frac{f}{w}$ = "flops per word moved"

q needs to be large to run fast:

$$f \cdot \gamma \geq w \cdot \beta \Rightarrow q = \frac{f}{w} \geq \frac{\beta}{\gamma} >> 1$$

History of how this has influenced algs.

BLAS-1 Basic Linear Algebra Subroutines

Standard Library of 15 ops mostly on vectors

(1) $y = \alpha \cdot x + y$ "axpy" x, y vectors, α scalar

inner loop of Gauss Elim

(2) dot product

(3) $\|x\|_2 = \sqrt{\sum_i |x_i|^2}$

(4) find largest entry $|x_i|$ in x

Motivation: easier programming, readable, avoid over/underflow, efficiency

Poor comp intensity: $q = \frac{2n}{2n} = 1$ (dot prod)

BLAS-2 (mid 1980s)

standard library of 25 ops on Matrix-vector pairs

(1) $y = \alpha y + \beta \cdot A \cdot x$ "GEMV" A matrix, x, y vecs, α, β scalars

lots of variations, $A = A^T$, triangular...

(2) $A = A + \alpha \cdot x \cdot y^T$ rank-one update
"GER"

(3) Solve $Tx = b$, T triangular "TRSV"

Motivation: similar to BLAS-1
+ opportunities to optimize on vector computers

Not much improvement on $q = \frac{f}{w} = \frac{2n^2}{n^2} = 2$
for GEMV $n \times n$

BLAS-3 library (late 1980s)
9 operations on pairs of matrices

(1) $C = \alpha C + B \cdot A \cdot B$ A, B, C matrices
"GEMM" α, B scalars

(2) $C = \alpha C + \beta \cdot A \cdot A^T$ $C = C^T$, A rectangular
"SYRK"

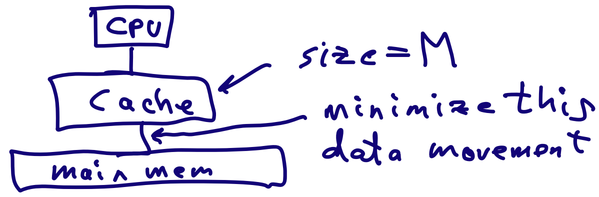
(3) Solve $TX = B$ T triangular, B matrix

q for GEMM = $\frac{f}{w} = \frac{2n^3}{4n^2} = \frac{n}{2}$ ($n \times n$)

But usual 3 nested loops as slow as GEMV

BLAS 3 led community to rewrite all dense linear algebra using BLAS
resulted in LAPACK, ScaLapack, ...

Goal: Prove comm lower bound for $n \times n$ matmul for



Easy case: all 3 matrices fit in cache: $3n^2 \le M$

read A, B, C into cache
do all arithmetic
write C back to main mem
words moved $= 4n^2$ obvious lower bound

Hard Case: $3n^2 > M$

Thm (Hong, Kung 1981): To multiply $C = C + A \cdot B$ (or $A \cdot B$) using $2n^3$ flops in any correct order, #words moved $= \Omega\left(\frac{n^3}{\sqrt{M}}\right)$

More modern proof: Irony, Tiskin, Toledo (2004)
Extends to rectangular, sparse matrices
$$\Omega\left(\frac{\#\text{flops}}{\sqrt{M}}\right)$$

proof Sketch:

1) Suppose we fill cache with M words
2) do as many flops as possible
3) store results back to main mem

Upper bound # flops we can do in 2) by G

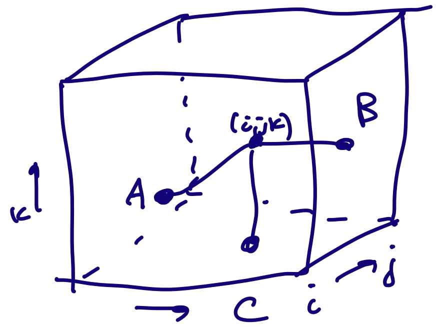
$\Rightarrow$ Doing G flops costs 2M words moved

$\Rightarrow$ since we do $2n^3$, need to repeat $\frac{2n^3}{G}$ times

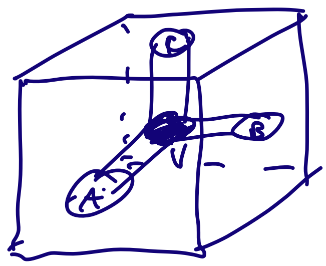
$\Rightarrow$ # words moved $= \frac{2n^3}{G} \cdot 2M$

Need G, or an upper bound

use geometric model: represent alg as 3D cube / lattice



for i=1:n
  for j=1:n
    for k=1:n
$C(i,j) \mathrel{+}= A(i,k) \cdot B(k,j)$



Goal: bound $|V|$ in terms

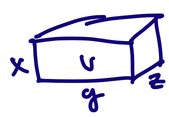
$|V_A| = \#A$ entries

$|V_B| = \#B$ entries

$|V_C| = \#C$ entries

assumption: $|V_A| + |V_B| + |V_C| \le M$

Intuition:



$|V| = x \cdot y \cdot z$

$|V_A| = x \cdot y$

$|V_C| = y \cdot z$

$|V_B| = x \cdot z$

$$|V| = \sqrt{|V_A| \cdot |V_B| \cdot |V_C|}$$

Thm (Loomis + Whitney 1949)

$$|V| \le \sqrt{|V_A| \cdot |V_B| \cdot |V_C|} \text{ for any } V$$

$$G = |V| \le \sqrt{M \cdot M \cdot M} = M^{3/2}$$

words moved $\geq \frac{2n^3}{G} \cdot 2M = \frac{4n^3 M}{M^{3/2}} = \Omega\left(\frac{n^3}{\sqrt{M}}\right)$

If careful with constants:

words moved $\geq \frac{2n^3}{\sqrt{M}} - 2M$

attainable!