Welcome to Ma221! (Feb 8)

Last time: "optimal" matmul

minimized comm by blocking,

depending on M = cache size

Problems: depends on M, # levels in cache

Eg 2 levels of cache $\Rightarrow$ 9 nested loops

Goal: optimal matmul independent of HW

use recursion:

function $C = RMM(A,B)$

... RMM = Recursive Matmul

... Simplicity: assume $A^{n \times n}$, $B^{n \times n}$

... and $n = 2^m$

... $C = \begin{array}{cc} n/2 & n/2 \end{array}$
$$C = \begin{array}{c} \\ \left[\begin{array}{cc} C_{11} & C_{12} \\ C_{21} & C_{22} \end{array}\right] \end{array} \begin{array}{l} n/2 \\ n/2 \end{array}$$

... if $n=1$ $C = A \cdot B$ else

... $C_{11} = RMM(A_{11}, B_{11}) + RMM(A_{12}, B_{21})$

.... ditto for C_{12}, C_{21}, C_{22}

Correctness by induction

Cost analysis:

$A(n) = \#$ arithmetic ops

$$= 8A\left(\frac{n}{2}\right) + 4\left(\frac{n}{2}\right)^2, \quad A(1) = 1$$

Solve $n = 2^m$

$$\frac{a(m)}{8^m} = \frac{A(2^m)}{8^m} = \frac{8 \cdot a(m-1) + 2^{2m}}{8^m}$$

$$b(m) = \frac{a(m)}{8^m} = b(m-1) + \frac{1}{2^m}$$

geometric sum

$W(n) = \#$ words moved

$$= 8W\left(\frac{n}{2}\right) + 4 \cdot 3 \cdot \left(\frac{n}{2}\right)^2$$

$$= 8W\left(\frac{n}{2}\right) + 3n^2$$

Base case when all 3 matrices fit in cache

$$W(b) = 3b^2 \text{ if } 3b^2 \le M$$

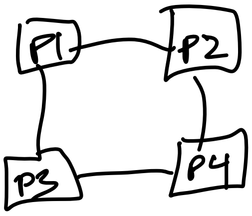
$$b = \sqrt{M/3}$$

Still geometric sum: $W(n) = O\left(\frac{n^3}{\sqrt{M}}\right)$

"Cache Oblivious": works for any cache size, any # levels of cache

works for much of linear algebra

Extension to parallel case



want to minimize comm over network

each processor does $\frac{1}{p}$ of flops "load balanced"

each processor store $\frac{1}{p}$ of all data

"fast memory" = local to each proc

"slow memory" = all rest on other procs

same lower bound $= \Omega\left(\frac{\#\text{flops per proc}}{\sqrt{\text{mem per proc}}}\right)$

$$= \Omega\left(\frac{n^3/p}{\sqrt{3n^2/p}}\right) = \Omega\left(\frac{n^2}{\sqrt{p}}\right)$$

attainable by parallel algs (SUMMA)

Going faster than $O(n^3)$ flops

Strassen (1967): matmul possible in $O(n^{\log_2 7})$ flops $\cong O(n^{2.81})$

trick: recursion with 7 calls, not 8

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

$$\left.\begin{aligned} P1 &= (A_{12} - A_{22}) \cdot (B_{21} + B_{22}) \\ P2 &= (A_{11} + A_{22}) \cdot (B_{11} + B_{22}) \\ &\vdots \\ P7 &= \end{aligned}\right) \begin{array}{l} \text{7 calls to} \\ \text{Strassen} \\ \text{(Recursive)} \end{array}$$

$$\left.\begin{aligned} C_{11} &= P1 + P2 - P4 + P6 \\ C_{12} &= P_4 + P_5 \\ &\vdots \end{aligned}\right) + \begin{array}{l} \text{total of} \\ \text{18 add/sub} \\ \text{of } (\frac{n}{2}) \times (\frac{n}{2}) \\ \text{matrices} \end{array}$$

$$A(n) = 7A\left(\frac{n}{2}\right) + 18\left(\frac{n}{2}\right)^2$$

$$\downarrow$$

$$A(n) = O(n^{\log_2 7}) \qquad \omega = \log_2 7$$

$$W(n) = O\left(\frac{n^{\omega}}{M^{\omega/2 - 1}}\right)$$

Thm (2010) $W(n)$ attains lower bound

Thm (2015) Extends to all "Strassen-like" algs

Latest record for smallest exponent ω

Thm (2020, Alman Williams)

$$\omega = 2.3728596$$

(not practical)

Thm (2008): All linear algebra can be done in $O(n^\omega)$ ops and "stably"

Error Analysis for Strassen

Usual Matmul (Q1.10)

$$|fl(A \cdot B) - A \cdot B| \leq n\varepsilon\, |A| \cdot |B|$$

Strassen:

$$\| fl(AB) - A \cdot B \| = O(\varepsilon) \cdot \|A\| \cdot \|B\|$$

Gauss's trick for complex matmul

$$(A + iB) \cdot (C + iD)$$

$$T_1 = A \cdot C$$
$$T_2 = B \cdot D$$
$$T_3 = (A + B) \cdot (C + D)$$

$(A+iB)\cdot(C+iD) = (T_1 - T_2) + i(T_3 - T_1 - T_2)$

cost : 3 matmuls + 5 adds
vs : 4 matmuls + 2 adds
cost about $\frac{3}{4}$ of traditional alg.

Gaussian Elimination

Additional goals, on top of avoiding comm, $O(n^\omega)$ flops

: Backward stability: exact solution of $(A+E)\hat{x} = b+f$
$\frac{\|E\|}{\|A\|} = O(\varepsilon)$, $\frac{\|f\|}{\|b\|} = O(\varepsilon)$

: Exploit math. structure of A
A: symmetric, positive definite
"sparse" = depends on $\ll n^2$ parameters, so could have many 0 entries, or be dense but depend on few parameters.
Eg: Vandermonde Matrix
$V_{ij} = x_i^{j-1}$ given $[x_1, \ldots, x_n]$

$V \cdot c = b$: multiplying by V

= polynomial evaluation

solving $Vc = b$ for c:

polynomial interpolation

much lower cost

Seek Matrix Factorizations.

A = product of simple matrices

SVD: $A = U \cdot \Sigma V^T$

= orthog · diag · orthog

Gaussian Elim:

$A = P \cdot L \cdot U$

P = permutation

L lower triangular ("unit" $L_{ii} = 1$)

U upper "

Least Squares $A = QR$

= orthog · uppertriangular

Eigen $A = QTQ^T$ Q orthog

T uppertriangular

Def: Permutation Matrix:

: identity matrix with permuted rows

Facts! Let P, P_1, P_2 be perms.

- P has exactly one 1 in each row and each column
- $P \cdot X = X$ with permuted rows
- $X \cdot P = X$ with permuted cols
- $P_1 \cdot P_2 =$ permutation
- $P^{-1} = P^T$ ie P orthogonal

  proof: note that $(P^T P)_{ii} = 1$

  $\Rightarrow P^T P = I$
- $\det(P) = \pm 1$
- storing and multiply by P cheap

  (store indices of 1s, copy rows)

Thm: (LU decomposition)

Given any $m \times n$ full rank A $m \geq n$

$\exists$ $m \times m$ perm P

$m \times n$ unit lower triangular L

$L =$ $\quad L_{ii} = 1$

$n \times n$ nonsingular U:

$A = P \cdot L \cdot U$

Cor: A $n \times n$ nonsingular $\Rightarrow$

$\exists$ $n \times n$ perm P,

$n \times n$ unit lower Δ L

$n \times n$ nonsingular ∇ U

$A = P \cdot L \cdot U$

to solve $Ax = b$

(1) Factor $A = P \cdot L \cdot U$

(expensive part: #flops $= \frac{2}{3} n^3 + O(n^2)$)

(2) Solve $P \cdot L \cdot U \cdot x = b$ for $L \cdot U x = P^T b$

by permuting b, $O(n)$

(3) Solve $L \cdot U \cdot x = P^T b$ for $U \cdot x = L^{-1} \cdot P^T \cdot b$

by forward substitution

cost $= n^2$

(4) Solve $Ux = L^{-1} P^T b$ for $x = U^{-1} L^{-1} P^T b$

by back substitution, cost $= n^2$

Given another b' can solve $Ax' = b'$

in just $O(n^2)$ ops

Note: We do not compute A^{-1} and multiply by A^{-1} because

(1) $3x$ more expensive in dense case (can be $O(n^2)x$ more expensive in sparse case)

(2) not numerically stable

Proof of $A = P \cdot L \cdot U$ (Gauss. Elim)

If A full rank $\Rightarrow$ first col nonzero

$\Rightarrow \exists$ perm P s.t $(PA)(1,1) \neq 0$

$$PA = \begin{array}{c|cc} & 1 & n-1 \\ \hline 1 & A_{11} & A_{12} \\ m-1 & A_{21} & A_{22} \end{array} = \begin{array}{c|cc} & 1 & m-1 \\ \hline 1 & 1 & 0 \\ m-1 & \frac{A_{21}}{A_{11}} & I \end{array} \cdot \begin{array}{c|cc} & 1 & n-1 \\ \hline 1 & A_{11} & A_{12} \\ m-1 & 0 & \underbrace{A_{22} - A_{21} \cdot A_{12} / A_{11}}_{S = \text{Schur Complement}} \end{array}$$

A fullrank $\Rightarrow PA$ full rank

$\Rightarrow S$ full rank! otherwise $\exists x \neq 0 : Sx = 0$

and then $A \begin{bmatrix} -A_{12} \cdot x / A_{11} \\ x \end{bmatrix} = 0 \Rightarrow A$ not full rank

Simpler in square case

$0 \neq \det(A) = \pm \det(PA) = \pm \det(1^{st}\text{ factor}) \cdot \det(2^{nd}\text{ factor})$

$$= \pm 1 \cdot A_{11} \cdot \det(S)$$

Apply induction to $S = P' \cdot L' \cdot U'$

$$PA = \left[\begin{array}{c|c} 1 & 0 \\ \hline \frac{A_{21}}{A_{11}} & I \end{array}\right] \left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline 0 & P' \cdot L' \cdot U' \end{array}\right]$$

$$= \left[\begin{array}{c|c} 1 & 0 \\ \hline \frac{A_{21}}{A_{11}} & P' \cdot L' \end{array}\right] \cdot \left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline 0 & U' \end{array}\right]$$

$$= \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & P' \end{array}\right] \cdot \left[\begin{array}{c|c} 1 & 0 \\ \hline P'^T \frac{A_{21}}{A_{11}} & L' \end{array}\right] \left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline 0 & U' \end{array}\right]$$

$$A = P^T \underbrace{\left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & P' \end{array}\right]}_{\text{perm}} \cdot \underbrace{\left[\begin{array}{c|c} 1 & 0 \\ \hline P'^T \frac{A_{21}}{A_{11}} & L' \end{array}\right]}_{\text{unit } \Delta} \cdot \underbrace{\left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline 0 & U' \end{array}\right]}_{\nabla}$$

QED

Express induction proof as Gauss. Elim (GE)

for $i = 1$ to n

... when $i = 1$ perform alg in proof

... when $i > 1$ apply same alg. to S

for $i=1$ to n

$L(i,i)=1$, $L(i+1:n,1)=A(i+1:n,i)/A(i,i)$

... ignore perms for now

$U(i,i:n)=A(i,i:n)$

if $(i<n)$ $A(i+1:n,i+1:n)=A(i+1:n,i+1:n)$
$- L(i+1:n,i)\cdot U(i,i+1:n)$

Add permutations: after "for $i=1$ to n", add

if $A(i,i)=0$ and some $A(j,i)\neq$ for $j>i$

swap rows i and j of L of A,

record swap in P

... how to choose $A(j,i)$ called "pivoting", details later

Don't waste space: let L and U overwrite A

row i of U overwrites row i of A:

omit $U(i,i:n)=A(i,i:n)$

col i of L (below diagonal) overwrites same entries of A, available because zeroed out: change first line to

$A(i+1:n,i)=A(i+1:n,i)/A(i,i)$

iterate from i to $n-1$

change last line to

$$A(i+1:n, i+1:n) = A(i+1:n, i+1:n) - A(i+1:n, i) \cdot A(i, i+1:n)$$

Summary:

for $i = 1$ to $n-1$

if $A(i,i) = 0 \neq A(j,i)$ for $j > i$,

swap rows i & j of A,

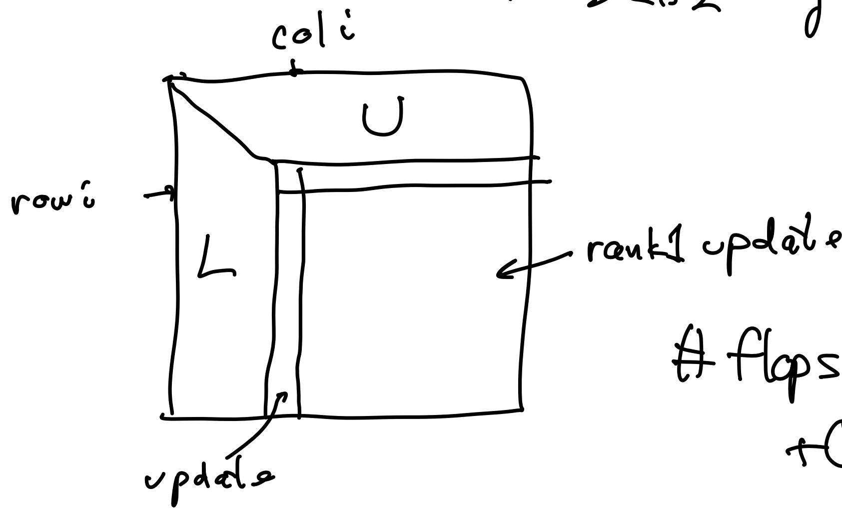
record swap in P

$A(i+1:n, i) = A(i+1:n, i) / A(i,i)$

... BLAS1 scal

$A(i+1:n, i+1:n) = A(i+1:n, i+1:n) - A(i+1:n, i) \cdot A(i, i+1:n)$

-- BLAS2 ger



$\#$ flops $= \frac{2}{3} n^3 + O(n^2)$