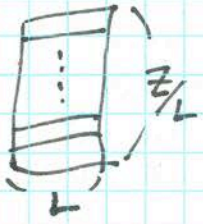
Note on Cache Oblivious Algorithms, CSE 549T

Chao Wang

- ideal cache model : (Z, L) = (size of cache, length of a cache line)

  three assumptions { ① tall cache : $Z = \Omega(L^2)$
  ② fully associative
  ③ optimal replacement policy (maintain perfect temporal locality)



- two complexities : ① work complexity $W(n)$ # of runtime
  ✓② cache complexity $Q(n; Z, L)$ # of cache misses

- motivating example : ~~se~~ matrix multiplication

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \times \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 1\cdot5+2\cdot7 & 1\cdot6+2\cdot8 \\ 3\cdot5+4\cdot7 & 3\cdot6+4\cdot8 \end{bmatrix} = \begin{bmatrix} 19 & 22 \\ 43 & 50 \end{bmatrix}$$

  to reduce # of cache misses ⟶ cache-aware algorithms (last week)
  ↳ need to include a ~~tuning~~ tuning parameter chosen based on the given cache layout.

- Cache Oblivious Algorithms

  (goal): achieve same Q as that of cache-aware algorithms without using any tuning parameters!

  (Examples): ① matrix multiplication
  ② matrix transposition

- ~~di~~ divide-n-conquer matrix multiplication algorithm

  (idea): keep halving the largest dimension of the two input matrixes A, B, and deal with each of the two subproblems.

$B' = [\frac{1}{2}]$
$A' = [\frac{1}{2}]$

  (Why): Although this algorithm may keep halving until the base case, at certain step all matrices will fit in cache and all induced subproblems will incur no cache misses

contd. three cases: ① $m \geq \max\{n,p\}$ $\begin{pmatrix} A_1 \\ A_2 \end{pmatrix} B = \begin{pmatrix} A_1 B \\ A_2 B \end{pmatrix}$

② $n \geq \max\{m,p\}$ $[A_1, A_2]\begin{bmatrix} B_1 \\ B_2 \end{bmatrix} = A_1B_1 + A_2B_2$

③ $p \geq \max\{m,n\}$ $A[B_1, B_2] = [AB_1 \;\; AB_2]$

Thm 1 proof (only talk about cases 1 and 3)

Reason for the bound on submatrices $\max\{m', n', p'\} \leq \alpha\sqrt{Z}$ ← the dimension of

↑ for analysis only, alg will keep dividing matrices until single-element matrix.

→ for all three submatrices fit into cache, we need $\begin{cases} m'n' = O(Z) \\ n'p' = O(Z) \\ m'p' = O(Z) \end{cases}$

$\Rightarrow m', n', p' = O(\sqrt{Z})$

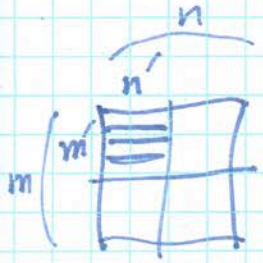
Case 1: $m, n, p > \alpha\sqrt{Z}$

$$Q(m,n,p) \leq \begin{cases} 2Q(m/2, n, p) + O(1), & m \text{ max} \\ 2Q(m, n/2, p) + O(1), & n \text{ max} \\ 2Q(m, n, p/2) + O(1), & p \text{ max} \\ \Theta((mn+np+mp)/L), & m,n,p \in [\frac{\alpha\sqrt{Z}}{2}, \alpha\sqrt{Z}] \end{cases}$$

when base case $m', n', p' \in [\frac{\alpha\sqrt{Z}}{2}, \alpha\sqrt{Z}] = \Theta(\sqrt{Z})$

$$Q(m',n',p') = \Theta\left(\frac{m'n'}{L} + m' + \frac{n'p'}{L} + n' + \frac{m'p'}{L} + m'\right) \quad (\text{recall } Z = \Omega(L^2),\ L = O(\sqrt{Z}))$$

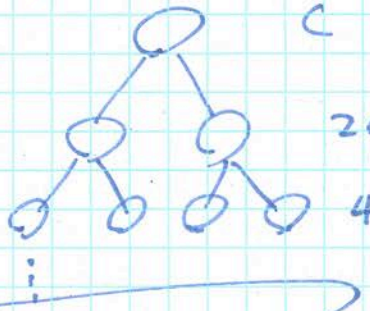
$$= \Theta((m'n' + n'p' + m'p')/L)$$



to compute $Q(m,n,p)$,

1 row → $\frac{n'}{L} + 1$ misses

m' rows → $m'(\frac{n'}{L} + 1) \leq \frac{m'n'}{L} + m'$

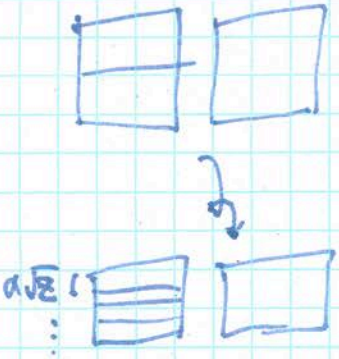


total Q at leaves dominates!

...d. therefore, $Q(m,n,p)$

$= \Theta($ # of leaves $\times$ Q at each leaf $)$

↳ i.e., the base case



$$\# \text{ of leaves} = \Theta\left(\frac{m}{\sqrt{z}}\right)\times\Theta\left(\frac{n}{\sqrt{z}}\right)\times\Theta\left(\frac{p}{\sqrt{z}}\right)$$

$$Q(m,n,p) = \Theta\left(\frac{mnp}{(\sqrt{z})^3}\times\frac{m'n'+n'p'+p'm'}{L}\right)$$

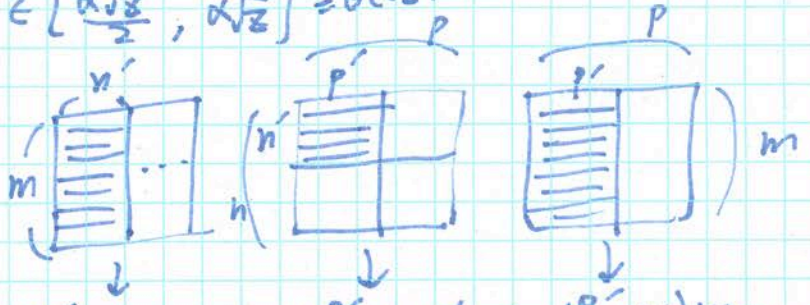
$$= \Theta\left(\frac{mnp}{L\sqrt{z}}\right) \quad *$$

<u>case 2</u> : $m \le \alpha\sqrt{z}$ $\quad n,p > \alpha\sqrt{z}$

$\Rightarrow$ we will keep doing $\begin{cases} (A_1\ A_2)\begin{pmatrix}B_1\\B_2\end{pmatrix} = A_1B_1 + A_2B_2 \\ \text{or } A(B_1\ B_2) = (AB_1\ \ AB_2) \end{cases}$

$\Rightarrow$ when $n', p' \in \left[\frac{\alpha\sqrt{z}}{2}, \alpha\sqrt{z}\right] = \Theta(\sqrt{z})$

base case



$$Q(m',n',p') = \left(\frac{n'}{L}+1\right)m + \left(\frac{p'}{L}+1\right)n' + \left(\frac{p'}{L}+1\right)m$$

$$= m + n' + \frac{mn'}{L} + \frac{n'p'}{L} + \frac{mp'}{L}$$

$$= \Theta\left(m + n' + \frac{n'p'}{L}\right)$$

$\because m = O(\sqrt{z})$
$p', n' = \Theta(\sqrt{z})$

↖ the paper has a 1 here, but that's for the case $m,n > \alpha\sqrt{z}$ and $p \le \alpha\sqrt{z}$

$$Q(m,n,p) = \Theta\left(m+n'+\frac{n'p'}{L}\right) \times \underbrace{\#\text{ of leaves}}_{\to\ \Theta\left(\frac{n}{\sqrt{z}}\cdot\frac{p}{\sqrt{z}}\right)}$$

$$= \Theta\left(\frac{(m+n')np}{\sqrt{z}\cdot\sqrt{z}} + \frac{n'p'np}{L\cdot\sqrt{z}\cdot\sqrt{z}}\right)$$

if $m = \overset{\text{small-o}}{o}(z) \longrightarrow Q = \Theta\left(\frac{np}{L}\right)$ from $\Theta\left(\frac{np}{\sqrt{z}} + \frac{np}{L}\right)$

$= \Theta\left(\frac{np}{L}\times m\right)$

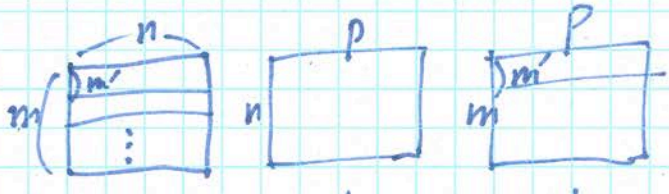
if $m = \Theta(z) \longrightarrow Q = \Theta\left(\frac{mnp}{L\sqrt{z}}\right)$

$\Rightarrow Q(m,n,p) = \Theta\left(\frac{np}{L} + \frac{mnp}{L\sqrt{z}}\right)$ *

case 3 : $n, p \leq \alpha\sqrt{z}$ $\quad m > \alpha\sqrt{z} = \Omega(\sqrt{z})$

$\Rightarrow$ we will keep doing $\begin{pmatrix} A_1 \\ A_2 \end{pmatrix} B = \begin{pmatrix} A_1 B \\ A_2 B \end{pmatrix}$

base case :



$m' = \Theta(\sqrt{z})$

$$Q(m', n, p) = \frac{nm'}{L} + 1 + \frac{np}{L} + 1 + \frac{pm'}{L} + 1$$

$$= \Theta\left(O(1) + \frac{nm'}{L} + \frac{pm'}{L}\right)$$

$$Q(m, n, p) = \Theta\left(\left(1 + \frac{nm'}{L} + \frac{pm'}{L}\right) \times \frac{m}{\sqrt{z}}\right)$$

$$= \Theta\left(\frac{m}{\sqrt{z}} + \frac{mnm'}{L\sqrt{z}} + \frac{mpm'}{L\sqrt{z}}\right)$$

if $n, p = \overset{\text{small-}O}{o}(\sqrt{z}) \Rightarrow \Theta\left(\frac{m}{\sqrt{z}} + \frac{mnm'}{L\sqrt{z}} + \frac{mpm'}{L\sqrt{z}}\right)$

$\because m = \Omega(\sqrt{z})$, $m' = \Theta(\sqrt{z})$

$$= \Theta(m)$$

if $n, p = \Theta(\sqrt{z}) \Rightarrow \Theta\left(\frac{m}{\sqrt{z}} + \frac{mnm'}{L\sqrt{z}} \cdot \frac{p}{p} + \frac{mpm'}{L\sqrt{z}} \cdot \frac{n}{n}\right)$

$$= \Theta\left(\frac{mnp}{L\sqrt{z}}\right)$$

dominate!

$$\Rightarrow Q(m, n, p) = \Theta\left(m + \frac{mnp}{L\sqrt{z}}\right)$$

- Matrix Transposition

$$\overset{A}{\underset{m}{\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{bmatrix}}} \overset{n}{\rightarrow} \overset{m}{\underset{n}{\begin{bmatrix} 1 & 5 \\ 2 & 6 \\ 3 & 7 \\ 4 & 8 \end{bmatrix}}}$$

in the context of this paper,
assume both A, B are in row major

```
Mat-Transpose(A, B, m, n)
  for i = 1 to m
    for j = 1 to n
      B_ji ← A_ij
```

huge A, B → every traversal of B_{ji} causes a miss

$\Rightarrow Q = \Theta(mn)$

contd. use divide-n-conquer

(idea): halving the larger dimension

$$m\begin{bmatrix} 1 & 2 & | & 3 & 4 \\ 5 & 6 & | & 7 & 8 \end{bmatrix}^{n} \rightsquigarrow \begin{bmatrix} 1 & 5 \\ 2 & 6 \\ \hline 3 & 7 \\ 4 & 8 \end{bmatrix} \Rightarrow [A_1\ A_2] \rightsquigarrow \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}$$

general pattern

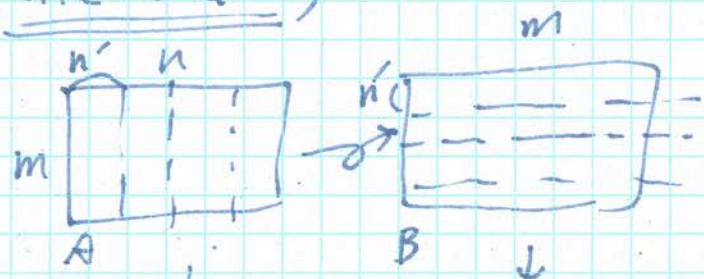
$$m\begin{bmatrix} 1 \\ 2 \\ \hline 3 \\ 4 \end{bmatrix}^{n} \rightsquigarrow [1\ 2 \mid 3\ 4] \Rightarrow \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} \rightsquigarrow [B_1\ B_2]$$

similarly to the matrix multiplication problem, there will be no further cache misses once both dimensions are small enough to fit in cache (whole matrix); let the threshold be αL

Three cases: none, one, or both dimensions are longer than αL

For the case $m \le \alpha L \le n$,

base:



$$\frac{\alpha L}{2} \le n' \le \alpha L$$

$$Q(m,n') = \left(\frac{n'}{L}+1\right)m + 1 + \frac{mn'}{L}$$

$$= 1 + m + 2\frac{mn'}{L} \ge 1 + m(1+\alpha) = O(1+m)$$

$$\left(\because n' \ge \frac{\alpha L}{2} \rightarrow \frac{2n'}{L} \ge \alpha\right)$$

$$Q(m,n) = \frac{n}{n'} \times O(1+m)$$

$$\ge \frac{n}{\alpha L} O(1+m) = O\left(\frac{n}{L} + \frac{mn}{L}\right) = O\left(\frac{mn}{L}\right) \quad (*)$$

~~$= O(1 + \text{since } L = \Omega(1))$~~

since $n = \Omega(L)$ $\quad = O\left(1 + \frac{mn}{L}\right)$ (*)