

Multi-threaded Algorithm 3

Michael Tsai

2014/1/2

Multithreaded matrix multiplication

P-SQUARE-MATRIX-MULTIPLY (A, B)

$$T_1(n) = \Theta(n^3)$$

`n=A.rows`

`let C be a new n x n matrix`

parallel `for i=1 to n`

parallel `for j=1 to n`

$$c_{ij} = 0$$

`for k=1 to n`

$$c_{ij} = c_{ij} + a_{ik} \cdot b_{kj}$$

`return C`

$$T_\infty(n) = \Theta(\log n) + \Theta(\log n) + \Theta(n) = \Theta(n)$$

$$\frac{T_1(n)}{T_\infty(n)} = \frac{\Theta(n^3)}{\Theta(n)} = \Theta(n^2)$$

Divide-and-conquer Multithreaded Algorithm for Matrix Multiplication (勒勒長)

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

$$\bullet B = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}$$

$$\bullet C = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}$$

$$= \begin{pmatrix} A_{11}B_{11} & A_{12}B_{12} \\ A_{21}B_{11} & A_{22}B_{12} \end{pmatrix} + \begin{pmatrix} A_{12}B_{21} & A_{12}B_{22} \\ A_{22}B_{21} & A_{22}B_{22} \end{pmatrix}$$

C
T

P-MATRIX-MULTIPLY-RECURSIVE (C, A, B)

n=A.rows

if n==1

$c_{11} = a_{11}b_{11}$

else let T be a new n x n matrix

partition A, B, C, and T into n/2 x n/2
submatrices $(A_{ij}, B_{ij}, C_{ij}, T_{ij}, i, j = \{1, 2\})$

spawn P-MATRIX-MULTIPLY-RECURSIVE (C₁₁, A₁₁, B₁₁)

spawn P-MATRIX-MULTIPLY-RECURSIVE (C₁₂, A₁₁, B₁₂)

spawn P-MATRIX-MULTIPLY-RECURSIVE (C₂₁, A₂₁, B₁₁)

spawn P-MATRIX-MULTIPLY-RECURSIVE (C₂₂, A₂₁, B₁₂)

spawn P-MATRIX-MULTIPLY-RECURSIVE (T₁₁, A₁₂, B₂₁)

spawn P-MATRIX-MULTIPLY-RECURSIVE (T₁₂, A₁₂, B₂₂)

spawn P-MATRIX-MULTIPLY-RECURSIVE (T₂₁, A₂₂, B₂₁)

P-MATRIX-MULTIPLY-RECURSIVE (T₂₂, A₂₂, B₂₂)

sync

parallel for i=1 to n

parallel for j=1 to n

$c_{ij} = c_{ij} + t_{ij}$

$$M_1(n) = 8M_1\left(\frac{n}{2}\right) + \Theta(n^2) = \Theta(n^3)$$

$$\frac{M_1(n)}{M_\infty(n)} = \frac{\Theta(n^3)}{\Theta(\log^2 n)} = \Theta\left(\frac{n^3}{\log^2 n}\right)$$

$$M_\infty(n) = M_\infty\left(\frac{n}{2}\right) + \Theta(\log n) + \Theta(\log n) = \Theta(\log^2 n)$$

How about Strassen's method?

- Reading assignment: p.795-796.
- Parallelism: $\Theta\left(\frac{n^{\log 7}}{\log^2 n}\right)$, slightly less than the original recursive version!

Multithreaded Merge Sort

Merge-Sort' (A, p, r)

if p < r

q = $\lfloor (p + r) / 2 \rfloor$

spawn MERGE-SORT' (A, p, q)

MERGE-SORT' (A, q+1, r)

sync

MERGE (A, p, q, r)

A: Array to be sorted

p and r: Start and end index of the range to be sorted

Merge() is the bottleneck!

$\Theta(n)$

$$MS'_1(n) = 2MS'_1\left(\frac{n}{2}\right) + \Theta(n) = \Theta(n \log n)$$

$$MS'_\infty(n) = MS'_\infty\left(\frac{n}{2}\right) + \Theta(n) = \Theta(n)$$

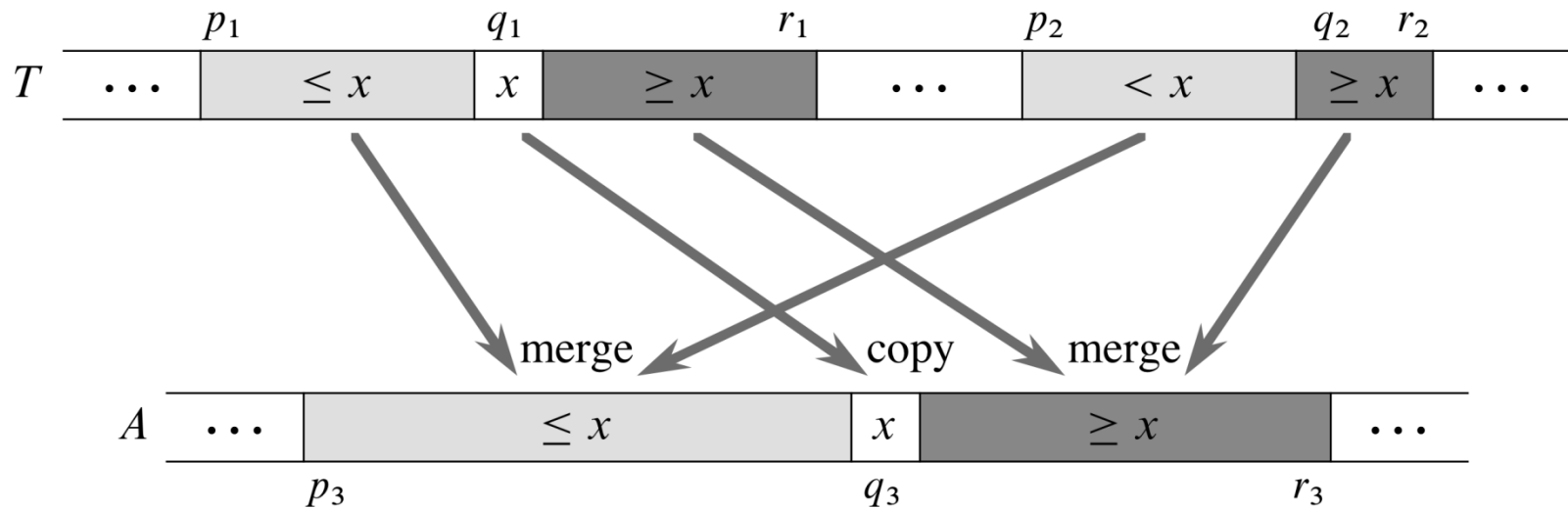
Parallelism = $\Theta(\log n)$



把Merge也用Divide & Conquer來解!
(方便交給不同的thread去做)

1. 挑出 $p_1 \sim r_1$ 的中位數 x

2. 找出 $p_2 \sim r_2$ 中 q_2 這個位置使得 $p_2 \sim q_2 - 1$ 都 $< x$, $q_2 \sim r_2$ 都 $\geq x$



3. Copy x 到新地方(根據 x 和 q_2 的位置)

4. 開分身去merge

$p_1 \sim q_1 - 1$ 和 $p_2 \sim q_2 - 1$, 及
 $q_1 + 1 \sim r_1$ 和 $q_2 \sim r_2$ 兩個區段

Base case:

$p_1 \sim r_1$ 和 $p_2 \sim r_2$ 都是空的

Multithreaded Merge

P-MERGE ($T, p_1, r_1, p_2, r_2, A, p_3$)

$n_1 = r_1 - p_1 + 1$

$n_2 = r_2 - p_2 + 1$

if $n_1 < n_2$

 exchange p_1 with p_2

 exchange r_1 with r_2

 exchange n_1 with n_2

if $n_1 == 0$

 return

else

$q_1 = \lfloor (p_1 + r_1) / 2 \rfloor$

$q_2 = \text{BINARY-SEARCH}(T[q_1], T, p_2, r_2)$

$q_3 = p_3 + (q_1 - p_1) + (q_2 - p_2)$

$A[q_3] = T[q_1]$

spawn P-MERGE ($T, p_1, q_1 - 1, p_2, q_2 - 1, A, p_3$)

 P-MERGE ($T, q_1 + 1, r_1, q_2, r_2, A, q_3 + 1$)

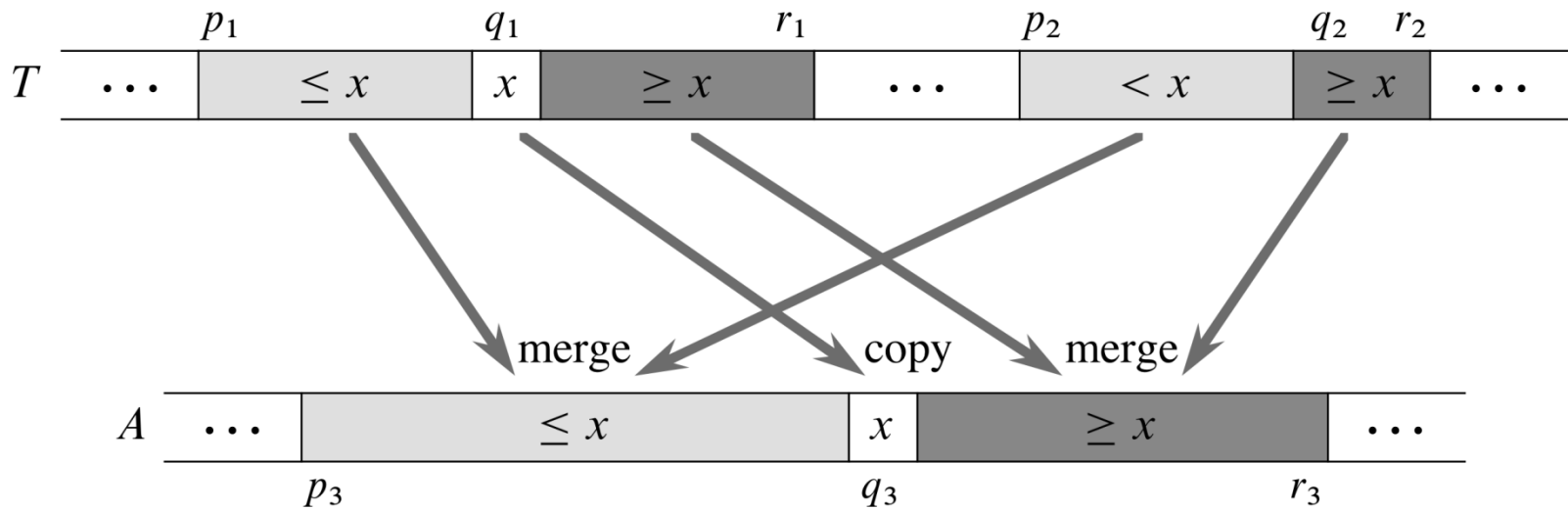
sync

T: Array to be merged

A: Array to save the merged result

p_1, r_1, p_2, r_2 : Start and end index of the range to be merged

p_3 : The index of the median in A



- $n_1 \geq n_2, n = n_1 + n_2$
- $n_2 = \frac{2n_2}{2} \leq \frac{n_1+n_2}{2} = \frac{n}{2}$
- 最糟的狀況下, x 比所有 $p_2 \sim r_2$ 的都大
- 要merge $\left\lfloor \frac{n_1}{2} \right\rfloor + n_2$ 個元素
- $\left\lfloor \frac{n_1}{2} \right\rfloor + n_2 \leq \frac{n_1}{2} + \frac{n_2}{2} + \frac{n_2}{2} = \frac{n_1+n_2}{2} + \frac{n_2}{2} \leq \frac{n}{2} + \frac{n}{4} = \frac{3n}{4}$

Multithreaded Merge

P-MERGE ($T, p_1, r_1, p_2, r_2, A, p_3$)

$n_1 = r_1 - p_1 + 1$

$n_2 = r_2 - p_2 + 1$

if $n_1 < n_2$

 exchange p_1 with p_2

 exchange r_1 with r_2

 exchange n_1 with n_2

if $n_1 == 0$

 return

else

$q_1 = \lfloor (p_1 + r_1) / 2 \rfloor$

$q_2 = \text{BINARY-SEARCH}(T[q_1], T, p_2, r_2)$ $\Theta(\log n)$

$q_3 = p_3 + (q_1 - p_1) + (q_2 - p_2)$

$A[q_3] = T[q_1]$

spawn P-MERGE ($T, p_1, q_1 - 1, p_2, q_2 - 1, A, p_3$)

 P-MERGE ($T, q_1 + 1, r_1, q_2, r_2, A, q_3 + 1$)

sync

$$PM_{\infty}(n) = PM_{\infty}\left(\frac{3n}{4}\right) + \Theta(\log n)$$

$$= \Theta(\log^2 n)$$

$PM_1(n) = \Omega(n)$, 因為至少要copy n elements

$PM_1(n) = O(n)$, 見課本p.802

$\Rightarrow PM_1(n) = \Theta(n)$

P-MERGE-SORT (A, p, r, B, s)

n=r-p+1

if n==1

B[s]=A[p]

else let T[1..n] be a new array

$$q = \left\lfloor \frac{p+r}{2} \right\rfloor$$

$$q' = q - p + 1$$

spawn P-MERGE-SORT (A, p, q, T, 1)

P-MERGE-SORT (A, q+1, r, T, q'+1)

sync

P-MERGE (T, 1, q', q'+1, n, B, s)

A: Array to be merged

p,r: Index of the range to be sorted

B: Array to save the result

$$PMS_1(n) = 2PMS_1\left(\frac{n}{2}\right) + PM_1(n) = 2PMS_1\left(\frac{n}{2}\right) + \Theta(n) = \Theta(n \log n)$$

$$PMS_\infty(n) = PMS_\infty\left(\frac{n}{2}\right) + PM_\infty(n) = PMS_\infty\left(\frac{n}{2}\right) + \Theta(\log^2 n) = \Theta(\log^3 n)$$

$$\text{Parallelism} = \Theta\left(\frac{n \log n}{\log^3 n}\right) = \Theta\left(\frac{n}{\log^2 n}\right)$$

Much better now!

怎麼自己學演算法

- 非常重要
- 這堂課沒辦法把所有“重要”的演算法都教完
- 但是希望過程中你已經建立了**自己學習演算法的能力**
- 準備研究所考試的時候可能會需要

我的經驗

- 我也跟著大家一起學習 資料結構 與 演算法
- 一些好方法(自以為)跟大家分享:
 1. 畫圖, 用簡單的例子圖解步驟
 2. 先了解概念(大方向), 不急著看複雜的數學 (我常說: 如果什麼都沒弄懂, 先弄懂**這個**)
 3. 一次了解**一小部分**就好 (模組化學習), 不急著弄懂所有的部分
 4. **相信自己**一定可以弄懂 (非常重要)
 5. 不知道自己到底懂了沒? 用一些**古怪**的例子來試試 (boundary case)
 6. 練習自己看課本 (非常重要)

期末考內容&型式

- Closed book, 2 x A4 cheat sheets (double-sided)
- 佔學期成績26%
- 包含整學期上課內容, 但以期中考過後的內容為主
- 180 minutes
- 題型與期中考類似 (是非+選擇+解釋)
- Extra office hour ??