1. A = , B =

Let *b11, b21, b31* is B1, *b12, b22, b32* is B2, and *b13, b23, b33* is B3. Then B = .

AB =

Let *a11, a12, a13* is A1, *a21, a22, a23* is A2, and *a31, a32, a33* is A3. Then A = .

AB = .

Let A =, A1 is m\*s, A2 is m\*(n-s), and B = , B1 is s\*n, B2 = (n-s)\*n.

Then .

If a matrix X = , and a matrix Y = , then if matrix C = XY, by above proof, we know that X1 = , X2 = , Y1 = , Y2 = , and

XY = => X1\*Y1 = = .

And so on, we can get C =

= .

1. Equation 1:

By the question, we know the number of additions is n3 and number of multiples is n3.

Equation 2:

M = n/q, and M is new columns of A (new rows of B). Similarly, N = n/p, and N is new row of A (new columns of B). For Cij, it will be added and multiplied M times, and for C, it exists N\*N element, so it need calculated N2 times.

For Cij, there are p\*p elements and for each element we need calculated M\*q (=n) times. Hence, for each Cij, the number of added and multiplied is p\*p\*n (n \* n2/N2). Totally, there are N2 elements, so the number of added is N2 \* n \* n2/N2 = n3, and the number of multiplied is N2 \* n \* n2/N2 = n3

(3)

執行情形：一張含有 電子用品, 螢幕, 鍵盤 的圖片

自動產生的描述

環境：

OS : macOS 10.15.5

CPU : 2.4GHz 4 cores intel Core i5

IDE : Xcode v12.0.1

(4)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| p q | 4 | 8 | 16 | 32 | 64 | 128 | 256 |
| 4 | 4.642643 | 4.033143 | 3.859925 | 3.865853 | 4.616160 | 5.022504 | 5.099792 |
| 8 | 3.655306 | 3.604086 | 3.503014 | 3.608221 | 4.483868 | 4.912957 | 5.043349 |
| 16 | 3.534409 | 3.464788 | 3.466462 | 3.606980 | 4.423390 | 4.880773 | 4.985149 |
| 32 | 3.586524 | 3.505883 | 3.542294 | 3.601915 | 4.396093 | 4.798906 | 4.897858 |
| 64 | 3.947464 | 3.542253 | 3.470878 | 3.575130 | 4.364373 | 4.769017 | 4.820594 |
| 128 | 3.970912 | 3.545517 | 3.459854 | 3.572529 | 4.352968 | 4.762901 | 4.888511 |
| 256 | 3.982609 | 3.540711 | 3.685007 | 3.917209 | 4.966024 | 5.064197 | 4.960946 |

(5)

a.

快取(cache)是一種暫存的記憶體。在CPU處理資料的時候會先從cache中找尋資料，如果從中有找到暫存的資料或指令，就不需要到main memory中找尋，因為CPU的處理速度大多比maim memory讀取速度快，使用cache較能有效的利用資源。此外，現今的cache還會分成L1、L2、L3等等，依據層級的不同，cache的儲存量及存取速度也都不同。

b. cache size of my computer.

一張含有 桌 的圖片

自動產生的描述

c. Explain the reason of the performance differences for various p and q.

因為matrix的乘法會不斷讀取到cache，但一般的matrix multiple無法有效的利用從cache拿出來的資料，造成時間的浪費，因此使用block matrix multiple。

以我的電腦來看cache L1 Data的size是32KB，而size of cache line size是64byte，也就是共有512個line，而一條line可以放16個int。因此可以看出來當p相同的時候，q是16的使用時間是最少的，這是因為可以一次拿出一條cache所有的資料並全數用完。當q大於16，就需要讀到第二條cache。而如果q小於16，同一條line則需要重複讀取，浪費掉了資源。