Cache: RAM Access

jueves, 7 de abril de 2022 06:55 a. m.

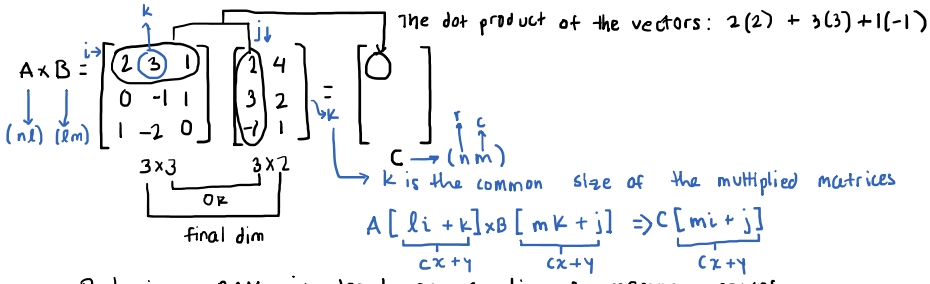
Parallel only: false sharing problem

↳ Due to locality principle, we are obliged to transfer whole blocks, and in order to have all words correct, we are constantly going to RAM: bad performance.

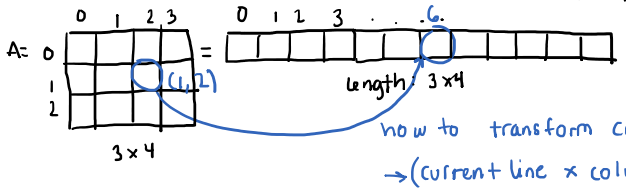
↳ the cause: cache is organized in lines

DevC++: Run thread-using programs:

Compiler options > Code Generation > Language standard: GNU C++11



But since RAM is ordered as a line of memory spaces



how to transform coordinates to line form?

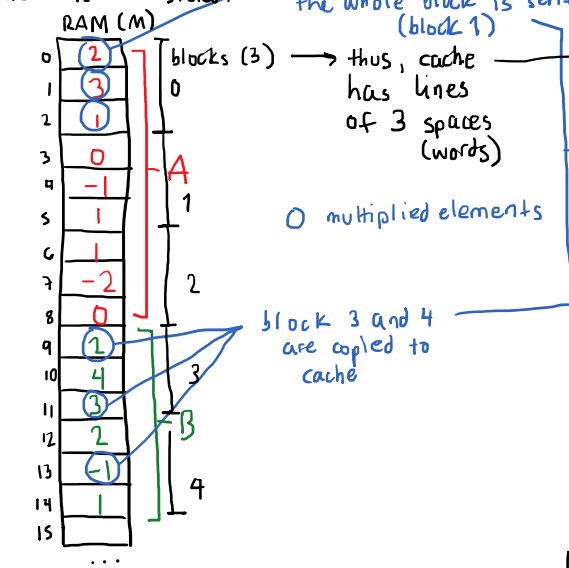
→ (current line x colums) + current column

we have: (1,2) = (x,y)

we want: (6) = 4x + y

↳ amount of columns

How is this stored?

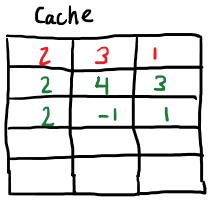


for this element, the whole block is sent (block 1)

thus, cache has lines of 3 spaces (words)

0 multiplied elements

block 3 and 4 are copied to cache



we would want that 2, 3, -1 is just one block: only 2 blocks in cache instead of 3

=

less copied RAM blocks to cache, less RAM access

=

better performance

how can we make B matrix elements contiguous?

↓

Transpose B!

$B = \begin{bmatrix} 2 & 3 & -1 \\ 4 & 2 & 1 \end{bmatrix}$ → together in RAM

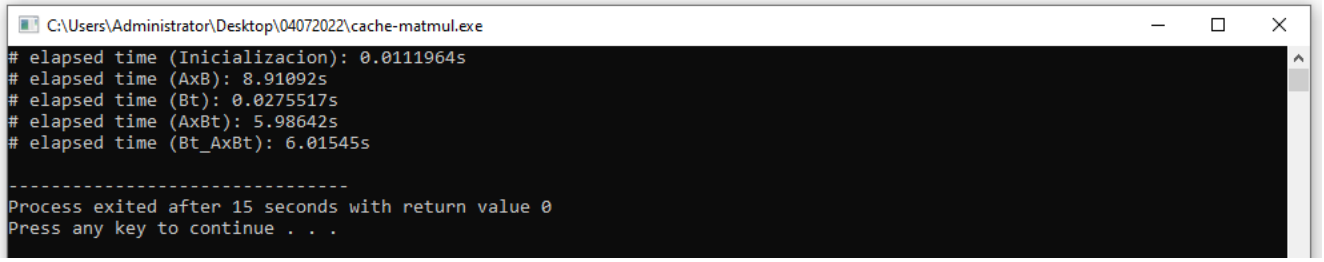
↳ but $A \times B \neq AB^t$

But we can change the code so that this is $AB^t = A \times B$

$$\begin{bmatrix} 2 & 3 & 1 \\ 0 & -1 & 1 \\ 1 & -2 & 0 \end{bmatrix} \begin{bmatrix} 2 & 3 & -1 \\ 4 & 2 & 1 \end{bmatrix} = \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix}$$

$A \times B^t = C$

Output



Even though this is a serial program, we see that there is a slower performance due to cache access. The parallel version would have serial parts, and this carry this problem.