CAP-387(2016) – Tópicos Especiais em Computação Aplicada: Construção de Aplicações Massivamente Paralelas

Aula 11: Caches e Blocagem

Celso L. Mendes, Stephan Stephany

LAC / INPE

Emails: celso.mendes@inpe.br, stephan.stephany@inpe.br

Caches – Relembrando...

- Organização das Caches
  - Vários níveis na hierarquia de memória (L1, L2, ...)
  - Níveis mais próximos à CPU têm menor *capacidade* e maior *velocidade*
  - Transferências de dados entre níveis são feitas por *blocos* (também chamados de *cache-lines* - exemplo: 64 Bytes)
  - Ao carregar uma linha na cache, pode ser necessário remover uma linha previamente carregada
    - Política comum de remoção: LRU (*Least-Recently Used*)
  - Operação de *store* pela CPU: escrita na memória (via cache)
    - *Write-through cache*: dado é escrito na cache e na memória; mais lento
    - *Write-back cache*: dado é escrito apenas na cache; mais rápido, porém memória pode ter que ser atualizada posteriormente

Caches – Santos Dumont

- CPU: Intel Ivy Bridge E5 2695 v2
  - 12 núcleos por chip, rodando a 2.4 GHz
  - L1 = 32 KB (individual por núcleo)
  - L2 = 256 KB (individual por núcleo)
  - L3 = 30 MB (compartilhada por todos os núcleos)
  - Documentação:
    - site da Intel
    - página na Wikipedia (procurar “Ivy Bridge-EP”)

Caches – Efeito em Programas

- Exemplo Ilustrativo: Transposição de Matrizes

```
DO J=1,N
        DO I=1,N
                B(I,J) = A(J,I)
        ENDDO
ENDDO
```

  - Assuma que A e B já foram inicializados e estão em memória
  - Número total de acessos a elementos de A e B: $2 \times N^2$
  - Loop **não** tem localidade temporal:
    - Cada element de A ou de B é acessado uma única vez
  - Dúvida: podem ocorrer efeitos de cache, em termos de desempenho?

Caches – Efeito em Programas

- Exemplo: Transposição de Matrizes (cont.)

```
DO J=1,N
        DO I=1,N
                B(I,J) = A(J,I)
        ENDDO
ENDDO
```

  - Taxa de acesso à memória: $2 \times N^2/T$ onde T é o tempo total dos loops
  - Execuções em um nó do Santos Dumont:

N	250	500	1.000	2.000	4.000
Taxa (MB/s)	1.655,6	1.231,5	169,4	141,3	107,1

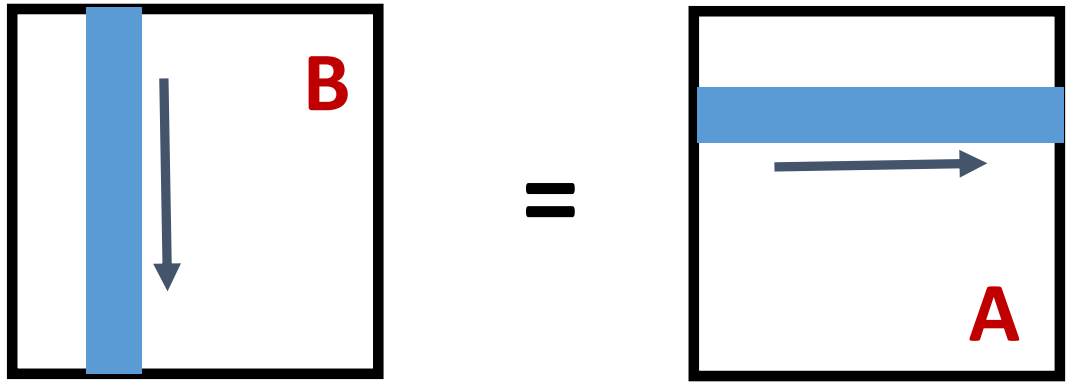
→ **???**

Caches – Efeito em Programas

- Exemplo: Transposição de Matrizes (cont.)

```
DO J=1,N
        DO I=1,N
                B(I,J) = A(J,I)
        ENDDO
ENDDO
```

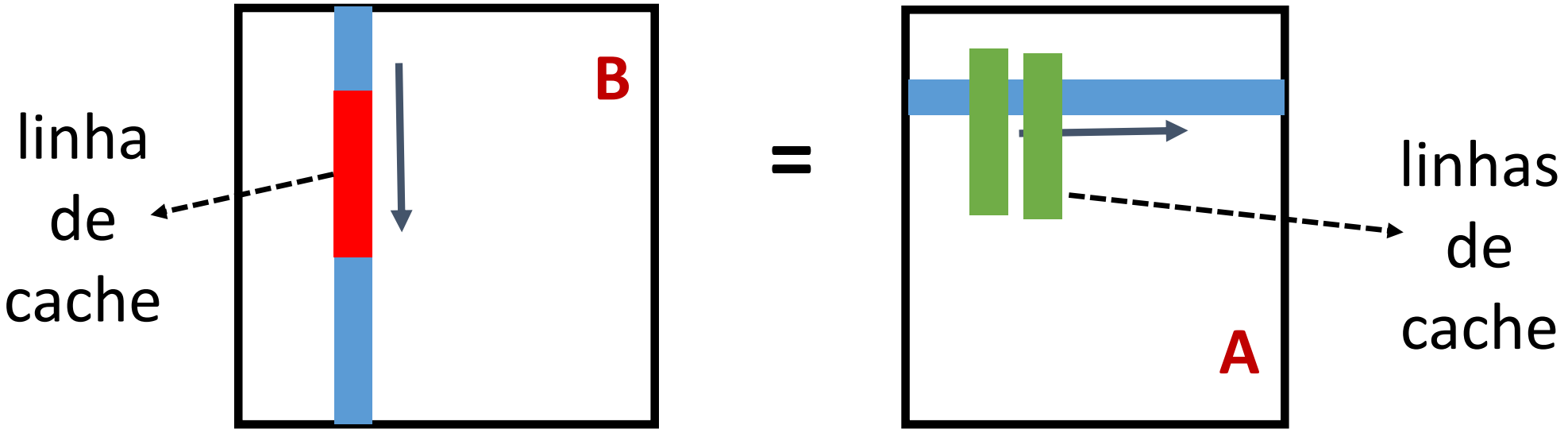
Acessos a B: endereços contíguos (Fortran)



Acessos a A: endereços com saltos de N posições

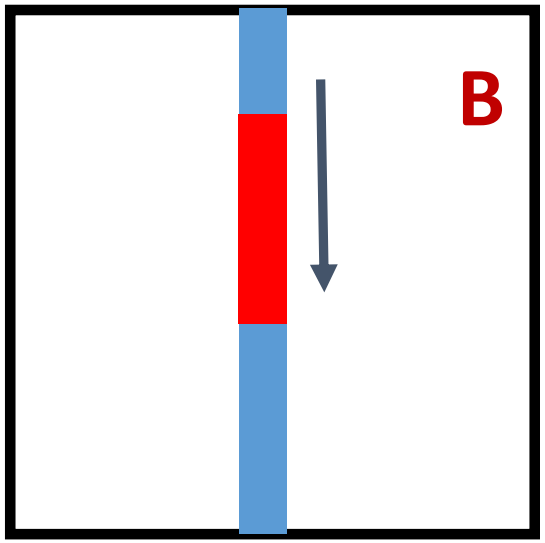
Caches – Efeito em Programas

- Exemplo: Transposição de Matrizes (cont.)
  - Acessos em termos de linhas de cache



Caches – Efeito em Programas

- Exemplo: Transposição de Matrizes (cont.)
  - Supondo que cada linha de cache tenha K palavras de 64 bits, isto é, K elementos de B:



A cada elemento B(I,J) acessado, ao ser trazido para a cache, outros K-1 elementos vizinhos também são trazidos!

→ Alta localidade espacial!

Caches – Efeito em Programas

- Exemplo: Transposição de Matrizes (cont.)
  - Por outro lado, com A os acessos são *diferentes*:

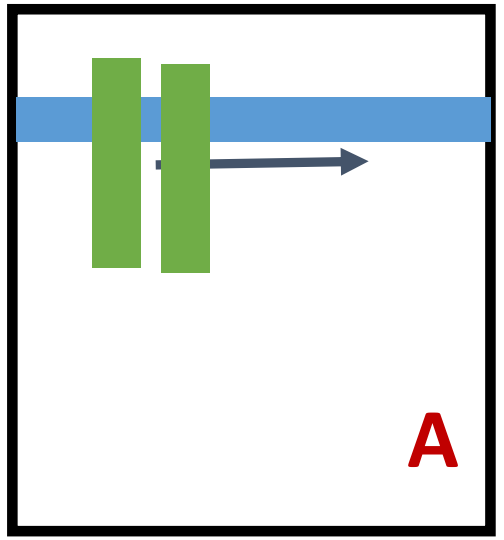
Ordem de acesso a A no loop:

A(J,I), A(J,I+1), A(J,I+2), …

Porém, os elementos numa linha são

A(J,I), A(J+1,I), A(J+2,I), …

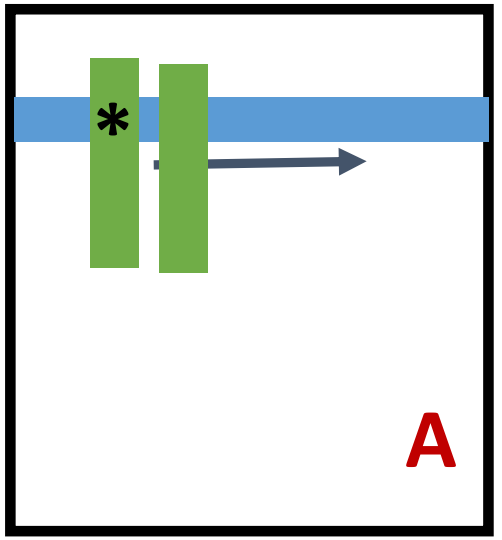
Logo, acessos a A(J,I) induzem a carga de vizinhos da mesma coluna!



Caches – Efeito em Programas

- Exemplo: Transposição de Matrizes (cont.)
  - Em outras palavras:

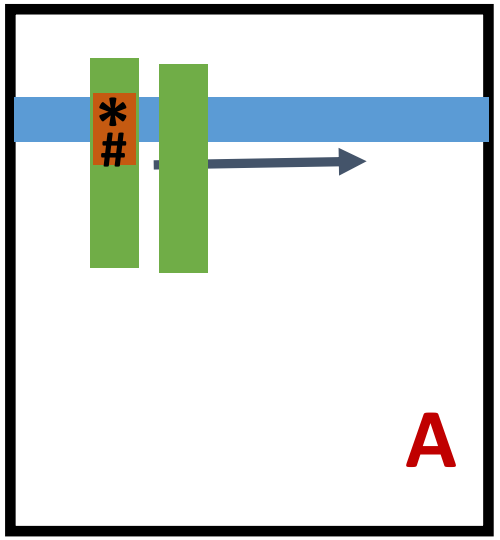
Para cada elemento A(J,I) acessado, ao ser trazido para a cache, outros K-1 elementos vizinhos (da mesma coluna) também serão trazidos!
→ Nenhum outro elemento da mesma linha de A (azul) é trazido para a cache (segmento verde)!



Caches – Efeito em Programas

- Exemplo: Transposição de Matrizes (cont.)
  - Única possibilidade de re-uso de dados de A em cache:

Ao acessar A(J,I)*, A(J+1,I)# será trazido para cache. Ele só será utlizado quando a próxima linha de A for varrida. Para que, até lá, A(J+1,I) ainda esteja em cache, a cache deve ser maior que ao menos uma linha da matriz A!

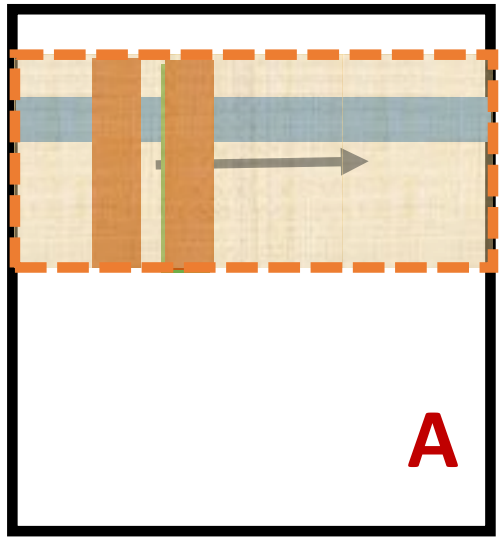


Caches – Efeito em Programas

- Exemplo: Transposição de Matrizes (cont.)
  - Condição geral para re-uso de todos os elementos de A:

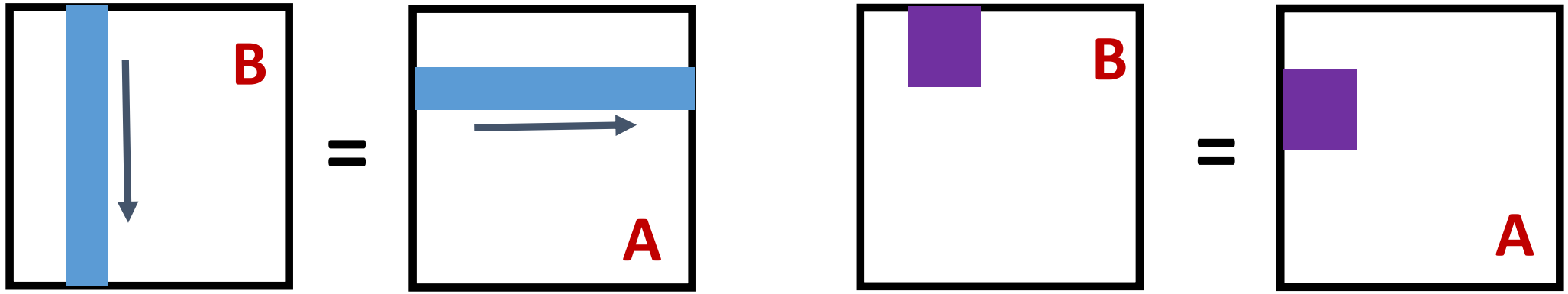
→Tamanho da linha de cache (em bytes) vezes N deve caber na cache!

Logo, o desempenho (isto é, a taxa efetiva de acesso a elementos de A) depend do tamanho do problema (N) e das características da cache!



Caches – Otimizações

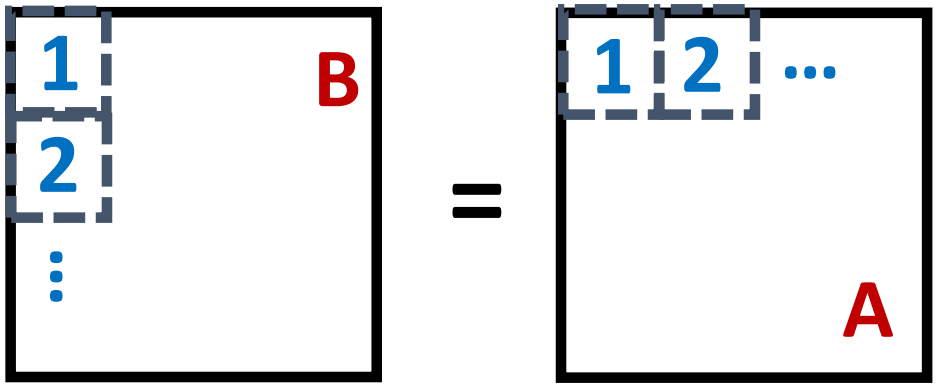
- Exemplo: Transposição de Matrizes (cont.)
  - Idéia principal para otimização do uso de cache:
    - Aumentar a probabilidade de re-uso dos valores de A



  - Novo esquema de execução:
    - Mudar o percorrimentodas das matrizes
    - Explorar o fato do loop original ter N^2 iterações independentes

Caches – Otimizações

- Exemplo: Transposição de Matrizes (cont.)
  - Novo esquema de execução: Blocagem de A e B
    - Dividir A e B em "azulejos" (*tiles*), operando em um de cada vez



→Tamanho dos azulejos:

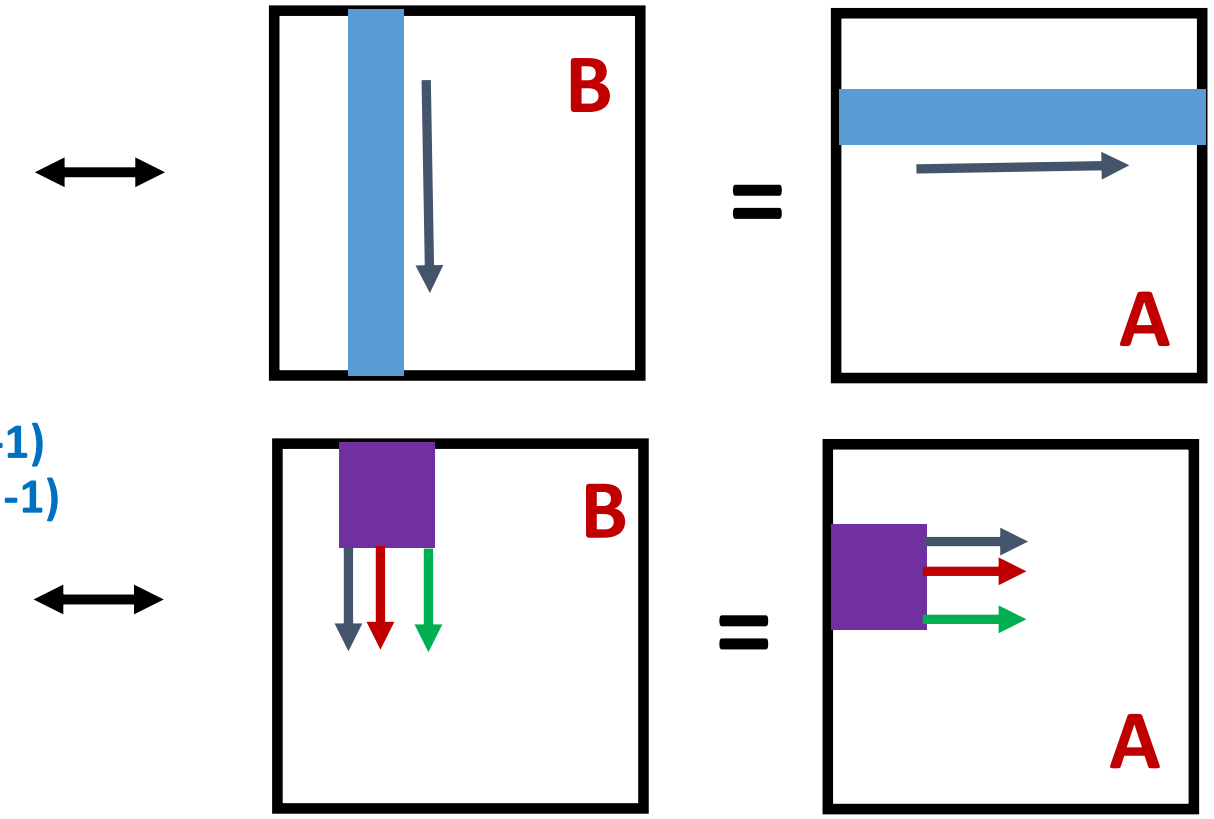
- Deve ser o maior possível (p/ minimizar o overhead dos novos loops)
- Deve ser pequeno o bastante para caber na cache do sistema

Caches – Blocagem

- Códigos: Original e Transformado

```
DO J=1,N
        DO I=1,N
                B(I,J) = A(J,I)
        ENDDO
ENDDO
```

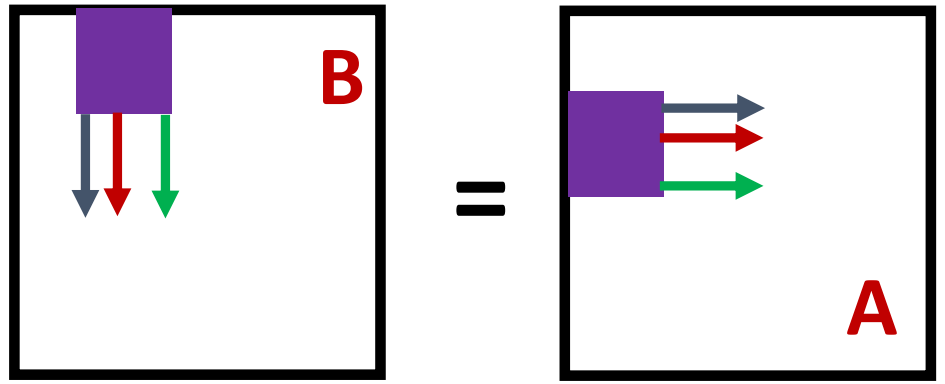
```
DO JJ=1,N,strideJ
  DO II=1,N,strideI
        DO J=JJ,min(N,JJ+strideJ-1)
          DO I=II,min(N,II+strideI-1)
                B(I,J) = A(J,I)
          ENDDO
        ENDDO
  ENDDO
ENDDO
```



Caches – Blocagem

- Código Transformado

```
DO JJ=1,N,strideJ
  DO II=1,N,strideI
        DO J=JJ,min(N,JJ+strideJ-1)
          DO I=II,min(N,II+strideI-1)
                B(I,J) = A(J,I)
          ENDDO
        ENDDO
  ENDDO
ENDDO
```



→ Quais os valores de *strideJ* e *strideI* ideais?

Caches – Blocagem

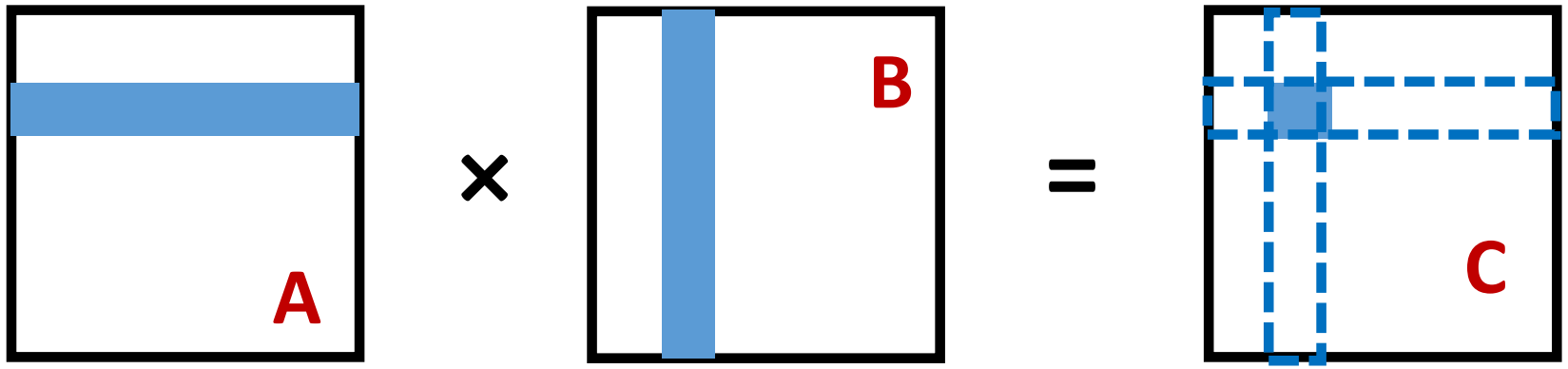
- Algumas reflexões sobre o exemplo:
  - Normalmente, *strideJ* = *strideI* = *stride* : ladrilhos quadrados
  - Escolha ideal do *stride*
    - Busca exaustiva: execuções com diversos valores distintos, busca do tempo mínimo
    - Geração de código e execuções automáticas, auto-tuning
  - Como fica isso num sistema com múltiplos níveis de cache?
  - Admitindo que *strideJ* ≠ *strideI*, qual deve ser maior?

Caches – Blocagem

- Muitas outras operações podem ter efeitos similares

  Exemplo: multiplicação de matrizes

  - Localidades temporal e especial, para as duas matrizes:



  - Linha de A é acessada múltiplas vezes, para gerar linha de C
  - Coluna de B é acessada múltiplas vezes, para gerar coluna de C
  - Acessos implicam em carga de linhas de cache, como na transposição