



MATRIZES

Prof. André Vignatti

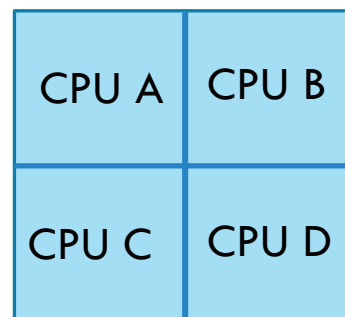
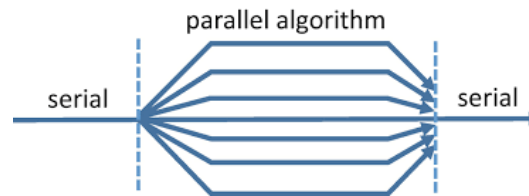
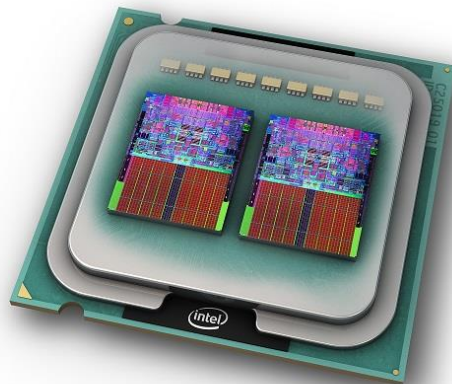
SOMA DE MATRIZES



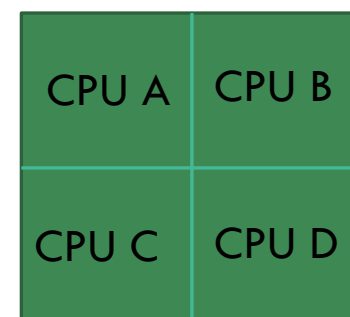
```
procedure somar_matrizes (var v, w, soma_v_w: matriz; n,m: integer);  
var i,j: integer;  
begin  
    (* n e m sao o numero de linhas e colunas, respectivamente *)  
    for i:= 1 to n do  
        for j:= 1 to m do  
            soma_v_w[i,j]:= v[i,j] + w[i,j];  
end;
```

SOMA DE MATRIZES

```
procedure somar_matrizes (var v, w, soma_v_w: matriz; n,m: integer);  
var i,j: integer;  
begin  
    (* n e m sao o numero de linhas e colunas, respectivamente *)  
    for i:= 1 to n do  
        for j:= 1 to m do  
            soma_v_w[i,j]:= v[i,j] + w[i,j];  
        end;  
    end;
```



+



esquema da GPU da Nvidia (2018)

INTRODUCING TURING

TU102 – FULL CONFIG

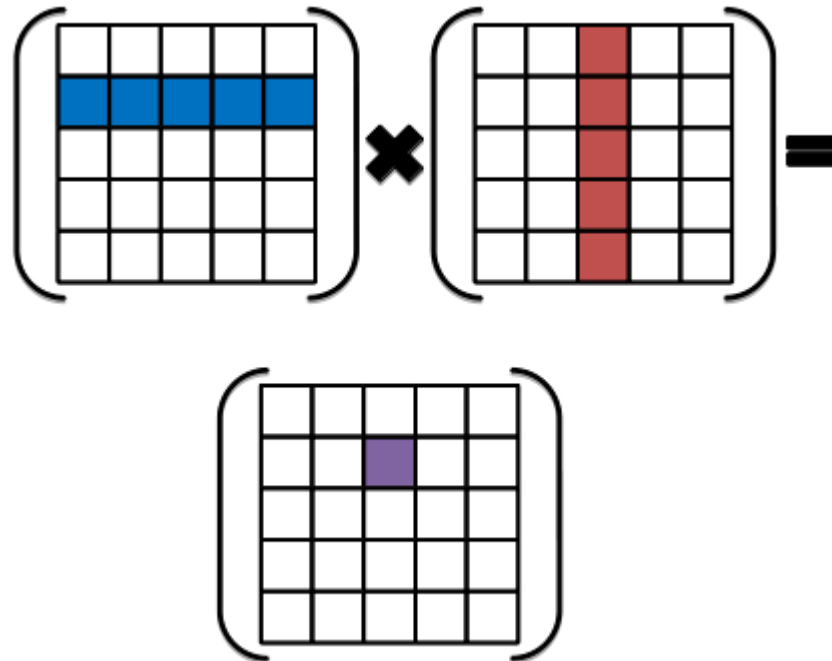
18.6 BILLION TRANSISTORS

SM	72
CUDA CORES	4608
TENSOR CORES	576
RT CORES	72
GEOMETRY UNITS	36
TEXTURE UNITS	288
ROP UNITS	96
MEMORY	384-bit 7 GHz GDDR6
NVLINK CHANNELS	2



MULTIPLICAÇÃO DE MATRIZES

Dadas duas matrizes A e B , queremos calcular $C = A \cdot B$



$$c_{ij} = a_{i1} \cdot b_{1j} + a_{i2} \cdot b_{2j} + a_{i3} \cdot b_{3j} + \cdots + a_{im} \cdot b_{mj} = \sum_{k=1}^m a_{ik} \cdot b_{kj}$$

MULTIPLICAÇÃO DE MATRIZES

Exemplo:

$$A: \begin{bmatrix} 4 & 6 & 2 & 1 \\ 9 & 0 & 0 & 2 \\ 8 & 7 & 3 & 9 \\ 1 & 2 & 3 & 4 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

$$B: \begin{bmatrix} 2 & 3 & 4 \\ 5 & 0 & 0 \\ 0 & 7 & 7 \\ 1 & 0 & 9 \end{bmatrix}$$

$$\begin{aligned} c_{32} &= a_{31} \cdot b_{12} + a_{32} \cdot b_{22} + a_{33} \cdot b_{32} + a_{34} \cdot b_{42} \\ &= 8 \cdot 3 + 7 \cdot 0 + 3 \cdot 7 + 9 \cdot 0 \\ &= 45 \end{aligned}$$

```
soma:= 0;
for i:= 1 to m do
    soma:= soma + A[I,k] * B[k,J];
prod_escalar:= soma;
```

MULTIPLICAÇÃO DE MATRIZES

É possível multiplicar quaisquer matrizes A e B ?

➤ somente se $A_{n \times m}$ e $B_{m \times l}$



MULTIPLICAÇÃO DE MATRIZES

```
procedure multiplicar_matrizes (var A: matriz;      lin_A, col_A: integer;
                                var B: matriz;      lin_B, col_B: integer;
                                var AB: matriz; var lin_AB, col_AB: integer);
var i, j, k: integer;
begin
  lin_AB:= lin_A; col_AB:= col_B;
  for i:= 1 to lin_A do
    for j:= 1 to col_B do
      begin
        AB[i,j]:= 0;
        for k:= 1 to lin_B do
          AB[i,j]:= AB[i,j] + A[i,k] * B[k,j];
        end;
      end;
    end;
  end;
```

tempo de execução: $O(n^3)$

CORRIDA PELO RECORDE

ano	execução	autores
1969	$O(n^{2.81})$	Strassen
1978	$O(n^{2.796})$	Pan
1979	$O(n^{2.78})$	Bini, Capovani, Romani, Lotti
1981	$O(n^{2.522})$	Schönhage
1981	$O(n^{2.517})$	Romani
1981	$O(n^{2.496})$	Coppersmith, Winograd
1986	$O(n^{2.479})$	Strassen
1990	$O(n^{2.376})$	Coppersmith, Winograd
2010	$O(n^{2.374})$	Stothers
2011	$O(n^{2.373})$	Williams
2014	$O(n^{2.372})$	Le Gall



Volker Strassen – vencedor do prêmio Knuth em 2008 pelo algoritmo seu algoritmo de 1969.

problema em aberto: há algoritmo $O(n^2)$?

ENCONTRANDO O MENOR ELEMENTO

```
function acha_menor_matriz (var w: matriz; n,m: integer): integer;  
var i,j: integer;  
    menor: integer;  
begin  
    menor:= w[1,1];  
    for i:= 1 to n do  
        for j:= 1 to m do  
            if w[i,j] < menor then  
                menor:= w[i,j];  
    acha_menor_matriz:= menor;  
end;
```

BUSCA EM MATRIZ

```
function busca (var w: matriz; n,m: integer; x: integer): boolean;
var i,j: integer; achou: boolean;
begin
    achou:= false;
    i:= 1;
    while (i <= n) and not achou do
    begin
        j:= 1;
        while (j <= m) and not achou do
        begin
            if w[i,j] = x then achou:= true;
            j:= j + 1;
        end;
        i:= i + 1;
    end;
    busca:= achou;
end;
```

BUSCA EM MATRIZ

```
function acha_pos_elemento (var w: matriz; n,m,x: integer; var l,c: integer): boolean;
var i,j: integer; achou: boolean;
begin
    achou:= false;
    i:= 1;
    while (i <= n) and not achou do
    begin
        j:= 1;
        while (j <= m) and not achou do
        begin
            if w[i,j] = x then
            begin
                (* quando acha o elemento , armazena as coordenadas *)
                achou:= true;
                l:= i;
                c:= j;
            end;
            j:= j + 1;
        end;
        i:= i + 1;
    end;
    acha_pos_elemento:= achou;
end;
```

ELEMENTOS REPETIDOS

```
function tem_repetidos (var w: matriz; n,m: integer): boolean;
var i,j, p, q: integer;
    repetiu: boolean;
begin
    repetiu:= false;
    i:= 1;
    while (i <= n) and not repetiu do
    begin
        j:= 1;
        while (j <= m) and not repetiu do
        begin
            p:= 1;
            while (p <= n) and not repetiu do
            begin
                q:= 1;
                while (q <= m) and not repetiu do
                begin
                    if (w[p,q] = w[i,j]) and ((p <> i) or (q <> j)) then
                        repetiu:= true;
                    q:= q + 1;
                end;
                p:= p + 1;
            end;
            j:= j + 1;
        end;
        i:= i + 1;
    end;
    tem_repetidos:= repetiu;
end;
```