

Récit d'un voyage en informatique, de 5 MFLOPS à 337 GFLOPS

Pierre-Évariste Dagand

IRIF – CNRS

5 Septembre 2022

#spoiler

Le calcul

Le calcul

$$\begin{pmatrix} 3 & 7 \\ 5 & 9 \end{pmatrix}$$

Le calcul

$$\begin{pmatrix} 2 & 8 \\ 6 & 4 \end{pmatrix}$$

$$\begin{pmatrix} 3 & 7 \\ 5 & 9 \end{pmatrix}$$

Le calcul

$$\begin{pmatrix} 2 & 8 \\ 6 & 4 \end{pmatrix} \begin{pmatrix} ? & ? \\ ? & ? \end{pmatrix} = \begin{pmatrix} 3 & 7 \\ 5 & 9 \end{pmatrix}$$

Le calcul

$$\begin{pmatrix} 3 & 7 \\ 5 & 9 \end{pmatrix}$$

$$\begin{pmatrix} 2 & 8 \\ 6 & 4 \end{pmatrix} \begin{pmatrix} (2 \times 3 + ?) & ? \\ ? & ? \end{pmatrix}$$

Le calcul

$$\begin{pmatrix} 3 & 7 \\ 5 & 9 \end{pmatrix}$$

$$\begin{pmatrix} 2 & 8 \\ 6 & 4 \end{pmatrix} \begin{pmatrix} (2 \times 3 + 8 \times 5) & ? \\ ? & ? \end{pmatrix}$$

Le calcul

$$\begin{pmatrix} 3 & 7 \\ 5 & 9 \end{pmatrix}$$

$$\begin{pmatrix} 2 & 8 \\ 6 & 4 \end{pmatrix} \begin{pmatrix} 46 & ? \\ ? & ? \end{pmatrix}$$

Le calcul

$$\begin{pmatrix} 3 & 7 \\ 5 & 9 \end{pmatrix}$$

$$\begin{pmatrix} 2 & 8 \\ 6 & 4 \end{pmatrix} \begin{pmatrix} 46 & (2 \times 7 + ?) \\ ? & ? \end{pmatrix}$$

Le calcul

$$\begin{pmatrix} 2 & 8 \\ 6 & 4 \end{pmatrix} \begin{pmatrix} 46 & (2 \times 7 + 8 \times 9) \\ ? & ? \end{pmatrix}$$
$$\begin{pmatrix} 3 & 7 \\ 5 & 9 \end{pmatrix}$$

Le calcul

$$\begin{pmatrix} 3 & 7 \\ 5 & 9 \end{pmatrix}$$

$$\begin{pmatrix} 2 & 8 \\ 6 & 4 \end{pmatrix} \begin{pmatrix} 46 & 86 \\ ? & ? \end{pmatrix}$$

Le calcul

$$\begin{pmatrix} 3 & 7 \\ 5 & 9 \end{pmatrix}$$

$$\begin{pmatrix} 2 & 8 \\ 6 & 4 \end{pmatrix} \begin{pmatrix} 46 & 86 \\ (6 \times 3 + 4 \times 5) & (6 \times 7 + 4 \times 9) \end{pmatrix}$$

Le calcul

$$\begin{pmatrix} 3 & 7 \\ 5 & 9 \end{pmatrix}$$

$$\begin{pmatrix} 2 & 8 \\ 6 & 4 \end{pmatrix} \begin{pmatrix} 46 & 86 \\ 38 & 78 \end{pmatrix}$$

Produit matriciel¹

En Mathématiques 1-4 (MI*)

Soient $\mathbf{A} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix}$ et $\mathbf{B} = \begin{pmatrix} b_{1,1} & b_{1,2} & \cdots & b_{1,p} \\ b_{2,1} & b_{2,2} & \cdots & b_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n,1} & b_{n,2} & \cdots & b_{n,p} \end{pmatrix}$

On définit la matrice produit $\mathbf{C} = \begin{pmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,p} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m,1} & c_{m,2} & \cdots & c_{m,p} \end{pmatrix}$

avec

$$c_{i,j} = a_{i,1}b_{1,j} + a_{i,2}b_{2,j} + \cdots + a_{i,n}b_{n,j} = \sum_{k=1}^n a_{i,k}b_{k,j}$$

1. “Matrix multiplication as composition”, 3Blue1Brown

Produit matriciel

En Informatique ?

Les matrices sont partout :

- Programmation réseaux (PR6)
- Bases de données (BD6)
- Cryptographie (CR6)
- Introduction à l'Apprentissage Profond (M2)
- ...

Produit matriciel

En Informatique ?

Les matrices sont partout :

- Programmation réseaux (PR6)
- Bases de données (BD6)
- Cryptographie (CR6)
- Introduction à l'Apprentissage Profond (M2)
- ...
- (Jeux vidéos!)

Produit matriciel

En Informatique ?



Jack Dongarra (Prix Turing 2021)

“Dongarra’s Algorithms and Software Fueled the Growth of High-Performance Computing and Had Significant Impacts in Many Areas of Computational Science from AI to Computer Graphics.”

Informatique de rue²

Temps d'exécution d'un produit matriciel?

Soient **A** et **B** deux matrices de tailles 4096×4096 .

1. Empreinte mémoire d'une matrice 4096×4096 ?

$4096 \times 4096 \times ?$ octets

2. *"Street-fighting mathematics", Mahajan*

Informatique de rue²

Temps d'exécution d'un produit matriciel?

Soient **A** et **B** deux matrices de tailles 4096×4096 .

1. Empreinte mémoire d'une matrice 4096×4096 ?

$$4096 \times 4096 \times 4 \text{ octets} = 2^? \times 2^? \times 2^2$$

2. *"Street-fighting mathematics", Mahajan*

Informatique de rue²

Temps d'exécution d'un produit matriciel?

Soient **A** et **B** deux matrices de tailles 4096×4096 .

1. Empreinte mémoire d'une matrice 4096×4096 ?

$$4096 \times 4096 \times 4 \text{ octets} = 2^{12} \times 2^{12} \times 2^2$$

2. *"Street-fighting mathematics", Mahajan*

Informatique de rue²

Temps d'exécution d'un produit matriciel?

Soient **A** et **B** deux matrices de tailles 4096×4096 .

1. Empreinte mémoire d'une matrice 4096×4096 ?

$$4096 \times 4096 \times 4 \text{ octets} = 2^{26} \text{ octets}$$

2. *"Street-fighting mathematics", Mahajan*

Informatique de rue²

Temps d'exécution d'un produit matriciel?

Soient **A** et **B** deux matrices de tailles 4096×4096 .

1. Empreinte mémoire d'une matrice 4096×4096 ?

$$4096 \times 4096 \times 4 \text{ octets} = 2^6 \text{ Mio}$$

2. *"Street-fighting mathematics", Mahajan*

Informatique de rue²

Temps d'exécution d'un produit matriciel?

Soient **A** et **B** deux matrices de tailles 4096×4096 .

1. Empreinte mémoire d'une matrice 4096×4096 ?

$$4096 \times 4096 \times 4 \text{ octets} = 64 \text{ Mio}$$

2. *"Street-fighting mathematics", Mahajan*

Informatique de rue²

Temps d'exécution d'un produit matriciel?

Soient **A** et **B** deux matrices de tailles 4096×4096 .

1. Empreinte mémoire d'une matrice 4096×4096 ? **64 Mio**
2. Nombre d'opérations flottantes scalaires?

$$\begin{pmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,4096} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,4096} \\ \vdots & \vdots & \ddots & \vdots \\ c_{4096,1} & c_{m,2} & \cdots & c_{4096,4096} \end{pmatrix} \quad \text{avec } c_{i,j} = \sum_{k=1}^{4096} a_{i,k} b_{k,j}$$

2. *"Street-fighting mathematics", Mahajan*

Informatique de rue²

Temps d'exécution d'un produit matriciel?

Soient **A** et **B** deux matrices de tailles 4096×4096 .

1. Empreinte mémoire d'une matrice 4096×4096 ? **64 Mio**
2. Nombre d'opérations flottantes scalaires?

$$\begin{pmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,4096} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,4096} \\ \vdots & \vdots & \ddots & \vdots \\ c_{4096,1} & c_{4096,2} & \cdots & c_{4096,4096} \end{pmatrix} \quad \text{avec } c_{i,j} = \sum_{k=1}^{4096} a_{i,k} b_{k,j}$$

$4096 \times 4096 \times ?$ FLOP

2. *"Street-fighting mathematics", Mahajan*

Informatique de rue²

Temps d'exécution d'un produit matriciel?

Soient **A** et **B** deux matrices de tailles 4096×4096 .

1. Empreinte mémoire d'une matrice 4096×4096 ? **64 Mio**
2. Nombre d'opérations flottantes scalaires?

$$\begin{pmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,4096} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,4096} \\ \vdots & \vdots & \ddots & \vdots \\ c_{4096,1} & c_{4096,2} & \cdots & c_{4096,4096} \end{pmatrix} \quad \text{avec } c_{i,j} = \sum_{k=1}^{4096} a_{i,k} b_{k,j}$$

$$4096 \times 4096 \times (4096 + 4096) \text{ FLOP}$$

2. *"Street-fighting mathematics", Mahajan*

Informatique de rue²

Temps d'exécution d'un produit matriciel?

Soient **A** et **B** deux matrices de tailles 4096×4096 .

1. Empreinte mémoire d'une matrice 4096×4096 ? **64 Mio**
2. Nombre d'opérations flottantes scalaires?

$$\begin{pmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,4096} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,4096} \\ \vdots & \vdots & \ddots & \vdots \\ c_{4096,1} & c_{m,2} & \cdots & c_{4096,4096} \end{pmatrix} \quad \text{avec } c_{i,j} = \sum_{k=1}^{4096} a_{i,k} b_{k,j}$$

$$2^{12} \times 2^{12} \times (2^{12} \times 2) \text{ FLOP}$$

2. *"Street-fighting mathematics", Mahajan*

Informatique de rue²

Temps d'exécution d'un produit matriciel?

Soient **A** et **B** deux matrices de tailles 4096×4096 .

1. Empreinte mémoire d'une matrice 4096×4096 ? **64 Mio**
2. Nombre d'opérations flottantes scalaires?

$$\begin{pmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,4096} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,4096} \\ \vdots & \vdots & \ddots & \vdots \\ c_{4096,1} & c_{m,2} & \cdots & c_{4096,4096} \end{pmatrix} \quad \text{avec } c_{i,j} = \sum_{k=1}^{4096} a_{i,k} b_{k,j}$$

2^{37} FLOP

2. *"Street-fighting mathematics", Mahajan*

Informatique de rue²

Temps d'exécution d'un produit matriciel?

Soient **A** et **B** deux matrices de tailles 4096×4096 .

1. Empreinte mémoire d'une matrice 4096×4096 ? **64 Mio**
2. Nombre d'opérations flottantes scalaires? **2^{37} FLOP**
3. Puissance crête de *c4.8xlarge*?

2. *"Street-fighting mathematics", Mahajan*

Informatique de rue²

Temps d'exécution d'un produit matriciel?

Soient **A** et **B** deux matrices de tailles 4096×4096 .

1. Empreinte mémoire d'une matrice 4096×4096 ? 64 Mio
2. Nombre d'opérations flottantes scalaires? 2^{37} FLOP
3. Puissance crête de *c4.8xlarge*?
 - Fréquence d'horloge : 2.9 GHz

2. *"Street-fighting mathematics"*, Mahajan

Informatique de rue²

Temps d'exécution d'un produit matriciel?

Soient **A** et **B** deux matrices de tailles 4096×4096 .

1. Empreinte mémoire d'une matrice 4096×4096 ? **64 Mio**
2. Nombre d'opérations flottantes scalaires? **2^{37} FLOP**
3. Puissance crête de *c4.8xlarge*?
 - Fréquence d'horloge : 2.9 GHz
 - Nombre d'opérations flottantes scalaires par cycle : 8 FLOP

2. *"Street-fighting mathematics", Mahajan*

Informatique de rue²

Temps d'exécution d'un produit matriciel?

Soient **A** et **B** deux matrices de tailles 4096×4096 .

1. Empreinte mémoire d'une matrice 4096×4096 ? **64 Mio**
2. Nombre d'opérations flottantes scalaires? **2^{37} FLOP**
3. Puissance crête de *c4.8xlarge*?
 - Fréquence d'horloge : 2.9 GHz
 - Nombre d'opérations flottantes scalaires par cycle : 8 FLOP
 - Nombre de coeurs : 36

2. *"Street-fighting mathematics", Mahajan*

Informatique de rue²

Temps d'exécution d'un produit matriciel?

Soient **A** et **B** deux matrices de tailles 4096×4096 .

1. Empreinte mémoire d'une matrice 4096×4096 ? 64 Mio
2. Nombre d'opérations flottantes scalaires? 2^{37} FLOP
3. Puissance crête de *c4.8xlarge*?
 - Fréquence d'horloge : 2.9 GHz
 - Nombre d'opérations flottantes scalaires par cycle : 8 FLOP
 - Nombre de coeurs : 36 $\Rightarrow 2.9 \times 8 \times 36$ GFLOPS

2. "*Street-fighting mathematics*", Mahajan

Informatique de rue²

Temps d'exécution d'un produit matriciel?

Soient **A** et **B** deux matrices de tailles 4096×4096 .

1. Empreinte mémoire d'une matrice 4096×4096 ? **64 Mio**
2. Nombre d'opérations flottantes scalaires? **2^{37} FLOP**
3. Puissance crête de *c4.8xlarge*?
 - Fréquence d'horloge : 2.9 GHz
 - Nombre d'opérations flottantes scalaires par cycle : 8 FLOP
 - Nombre de coeurs : 36 $\Rightarrow 2.9 \times 8 \times 36 \approx 2^2 \times 2^3 \times 2^5 \text{ GFLOPS}$

2. "*Street-fighting mathematics*", Mahajan

Informatique de rue²

Temps d'exécution d'un produit matriciel?

Soient **A** et **B** deux matrices de tailles 4096×4096 .

1. Empreinte mémoire d'une matrice 4096×4096 ? **64 Mio**
2. Nombre d'opérations flottantes scalaires? **2^{37} FLOP**
3. Puissance crête de *c4.8xlarge*?
 - Fréquence d'horloge : 2.9 GHz
 - Nombre d'opérations flottantes scalaires par cycle : 8 FLOP
 - Nombre de cœurs : 36 $\Rightarrow 2.9 \times 8 \times 36 \approx 2^{10}$ GFLOPS

2. *"Street-fighting mathematics", Mahajan*

Informatique de rue²

Temps d'exécution d'un produit matriciel?

Soient **A** et **B** deux matrices de tailles 4096×4096 .

1. Empreinte mémoire d'une matrice 4096×4096 ? 64 Mio
2. Nombre d'opérations flottantes scalaires? 2^{37} FLOP
3. Puissance crête de *c4.8xlarge*? $\approx 2^{10}$ GFLOPS
4. Temps d'exécution théorique?

2. "*Street-fighting mathematics*", Mahajan

Informatique de rue²

Temps d'exécution d'un produit matriciel?

Soient **A** et **B** deux matrices de tailles 4096×4096 .

1. Empreinte mémoire d'une matrice 4096×4096 ? **64 Mio**
2. Nombre d'opérations flottantes scalaires? **2^{37} FLOP**
3. Puissance crête de *$c4.8xlarge$* ? **$\approx 2^{10}$ GFLOPS**
4. Temps d'exécution théorique?

$$\frac{2^{37}}{2^{10} \times 10^9} \text{ secondes}$$

2. *"Street-fighting mathematics", Mahajan*

Informatique de rue²

Temps d'exécution d'un produit matriciel?

Soient **A** et **B** deux matrices de tailles 4096×4096 .

1. Empreinte mémoire d'une matrice 4096×4096 ? **64 Mio**
2. Nombre d'opérations flottantes scalaires? **2^{37} FLOP**
3. Puissance crête de *$c4.8xlarge$* ? **$\approx 2^{10}$ GFLOPS**
4. Temps d'exécution théorique?

$$\frac{2^{37}}{2^{10} \times 10^9} = 2^{27} \times 10^{-9} \text{ secondes}$$

2. *"Street-fighting mathematics", Mahajan*

Informatique de rue²

Temps d'exécution d'un produit matriciel?

Soient **A** et **B** deux matrices de tailles 4096×4096 .

1. Empreinte mémoire d'une matrice 4096×4096 ? 64 Mio
2. Nombre d'opérations flottantes scalaires? 2^{37} FLOP
3. Puissance crête de $c4.8xlarge$? $\approx 2^{10}$ GFLOPS
4. Temps d'exécution théorique?

$$\frac{2^{37}}{2^{10} \times 10^9} \approx 2^{27} \times 2^{-30} \text{ secondes}$$

2. *"Street-fighting mathematics", Mahajan*

Informatique de rue²

Temps d'exécution d'un produit matriciel?

Soient **A** et **B** deux matrices de tailles 4096×4096 .

1. Empreinte mémoire d'une matrice 4096×4096 ? **64 Mio**
2. Nombre d'opérations flottantes scalaires? **2^{37} FLOP**
3. Puissance crête de *$c4.8xlarge$* ? **$\approx 2^{10}$ GFLOPS**
4. Temps d'exécution théorique?

$$\frac{2^{37}}{2^{10} \times 10^9} \approx 2^{-3} \text{ secondes}$$

2. *"Street-fighting mathematics", Mahajan*

Informatique de rue²

Temps d'exécution d'un produit matriciel?

Soient **A** et **B** deux matrices de tailles 4096×4096 .

1. Empreinte mémoire d'une matrice 4096×4096 ? **64 Mio**
2. Nombre d'opérations flottantes scalaires? **2^{37} FLOP**
3. Puissance crête de *c4.8xlarge*? **$\approx 2^{10}$ GFLOPS**
4. Temps d'exécution théorique?

$$\frac{2^{37}}{2^{10} \times 10^9} \approx 1/8 \text{ secondes}$$

2. *"Street-fighting mathematics", Mahajan*

Informatique de rue²

Temps d'exécution d'un produit matriciel?

Soient **A** et **B** deux matrices de tailles 4096×4096 .

1. Empreinte mémoire d'une matrice 4096×4096 ? **64 Mio**
2. Nombre d'opérations flottantes scalaires? **2^{37} FLOP**
3. Puissance crête de *$c4.8xlarge$* ? **$\approx 2^{10}$ GFLOPS**
4. Temps d'exécution théorique?

$$\frac{2^{37}}{2^{10} \times 10^9} \approx 0.1 \text{ secondes}$$

2. *"Street-fighting mathematics", Mahajan*

Informatique de rue²

Temps d'exécution d'un produit matriciel?

Soient **A** et **B** deux matrices de tailles 4096×4096 .

1. Empreinte mémoire d'une matrice 4096×4096 ? 64 Mio
2. Nombre d'opérations flottantes scalaires? 2^{37} FLOP
3. Puissance crête de *c4.8xlarge*? $\approx 2^{10}$ GFLOPS
4. Temps d'exécution théorique? ≈ 0.1 s

2. "*Street-fighting mathematics*", Mahajan

Multiplication de matrice ? Trivial.

Essayons quand même³...

3. *"There's plenty of room at the Top", Leiserson et al.*

Version 1. Python

Après Initiation à la programmation 1-2 (IP*) et Langages de script (LS6)

```
n = 4096

def mm(A, B, C):
    for i in range(n):
        for j in range(n):
            for k in range(n):
                C[i][j] += A[i][k] * B[k][j]
```

Nombre de LoC : ≈ 10

Temps d'exécution : ?

Fraction de perf. crête : ?

Version 1. Python

Après Initiation à la programmation 1-2 (IP*) et Langages de script (LS6)

```
n = 4096

def mm(A, B, C):
    for i in range(n):
        for j in range(n):
            for k in range(n):
                C[i][j] += A[i][k] * B[k][j]
```

Nombre de LoC :	≈ 10
Temps d'exécution :	25 552,48 s (≈ 6 heures)
Fraction de perf. crête :	0.00 %

Version 1. Python

Éléments d'explication

```
$ cd 01_python; make
```


Version 1. Python

Éléments d'explication



Version 2. Java

Après Programmation orientée objets (POO3) et Machines virtuelles (MV6)

```
for (int i = 0; i < N; i++) {  
    for (int j = 0; j < N; j++) {  
        for (int k = 0; k < N; k++) {  
            C[i][j] += A[i][k] * B[k][j];  
        }  
    }  
}
```

Nombre de LoC :	≈ 30
Temps d'exécution :	2 372,68 s (≈ 40 minutes)
Accélération relative :	$\times 10.8$
Fraction de perf. crête :	0.01 %

Version 2. Java

Éléments d'explication

```
$ cd 02_java; make
```

Version 3. C

Après Langage C (LC4)



Dennis Ritchie & Ken Thompson⁴

4. Cf. *Aides à l'équipement U-Paris (barbes unisexe)*

Version 3. C

Après Langage C (LC4)

```
for (int i = 0; i < n; ++i) {  
    for (int j = 0; j < n; ++j) {  
        for (int k = 0; k < n; ++k) {  
            C[i][j] += A[i][k] * B[k][j];  
        }  
    }  
}
```

Version 3. C

Après Langage C (LC4)

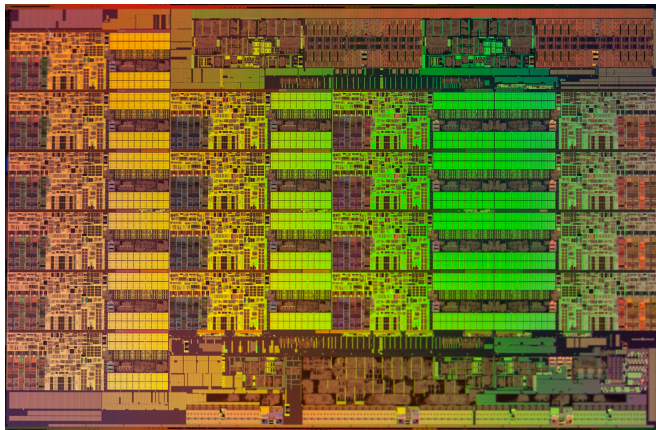
```
.L11:
    movapd    (%rdi), %xmm1
    movq      %rsi, %rax
    movq      %r8, %rdx

.L8:
    movsd     (%rdx), %xmm0
    addq      $32768, %rax
    addq      $8, %rdx
    unpcklpd  %xmm0, %xmm0
    mulpd     -32768(%rax), %xmm0
    addpd     %xmm0, %xmm1
    cmpq      %rcx, %rax
    jne       .L8
    addq      $16, %rsi
    movaps    %xmm1, (%rdi)
    leaq      16(%rax), %rcx
    addq      $16, %rdi
    cmpq      %r9, %rsi
    jne       .L11
```

Nombre de LoC : ≈ 40
Temps d'exécution : 542.67 s
Accélération relative : $\times 4.4$
Accélération absolue : $\times 47$
Fraction de perf. crête : 0.03 %

Version 3. C

Éléments d'explication



Intel Xeon E5-2666

Version 4. Boucles parallèles en *Cilk*

Après Systèmes d'exploitation (SY5)

$$\begin{pmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,4096} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,4096} \\ \vdots & \vdots & \ddots & \vdots \\ c_{4096,1} & c_{m,2} & \cdots & c_{4096,4096} \end{pmatrix} \quad \text{avec } c_{i,j} = \sum_{k=1}^{4096} a_{i,k} b_{k,j}$$

Version 4. Boucles parallèles en *Cilk*

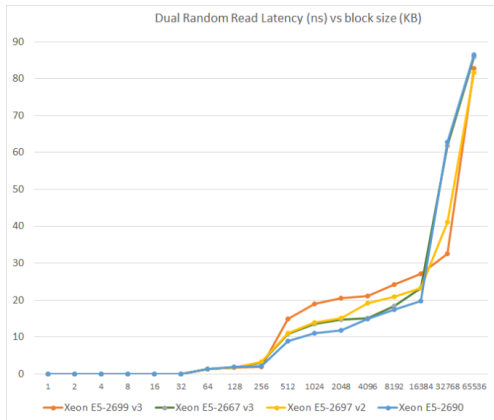
Après Systèmes d'exploitation (SY5)

```
cilk_for (int i = 0; i < n; ++i) {  
    cilk_for (int j = 0; j < n; ++j) {  
        for (int k = 0; k < n; ++k) {  
            C[i][j] += A[i][k] * B[k][j];  
        }  
    }  
}
```

Nombre de LoC :	≈ 40
Temps d'exécution :	69.80 s
Accélération relative :	$\times 7.8$
Accélération absolue :	$\times 366$
Fraction de perf. crête :	0.24 %

Version 4. Boucles parallèles en *Cilk*

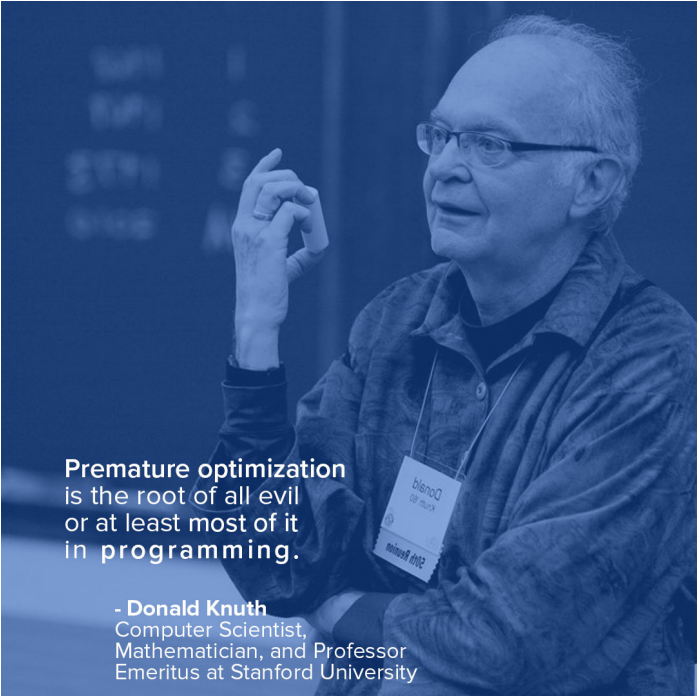
Éléments d'explication



Latence cache L1 : 0.5 ns

Latence cache L2 : 7 ns ($\times 14$ relatif)

Latence mémoire : 100 ns ($\times 20$ relatif, $\times 200$ absolu)

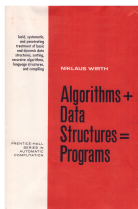
A photograph of Donald Knuth, an older man with glasses, wearing a dark jacket over a collared shirt. He is gesturing with his right hand, fingers curled, as if speaking. He has a conference badge hanging from his neck. The background is blurred, showing what appears to be a stage or lecture hall setting. The entire image has a blue color overlay.

Premature optimization
is the root of all evil
or at least most of it
in **programming**.

- **Donald Knuth**
Computer Scientist,
Mathematician, and Professor
Emeritus at Stanford University

Optimisation⁵

Après Éléments d'algorithmique 1-2 (EA3) et Programmation efficace (PE6)



Algorithmes : efficacité

Minimiser le nombre d'opérations à exécuter

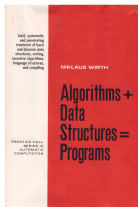
Structures de données : performance

Durée (physique) du calcul

5. *"Efficiency with Algorithms, Performance with Data Structures", Carruth*

Optimisation

Après Éléments d'algorithmique 1-2 (EA3) et Programmation efficace (PE6)



Algorithmes : efficacité

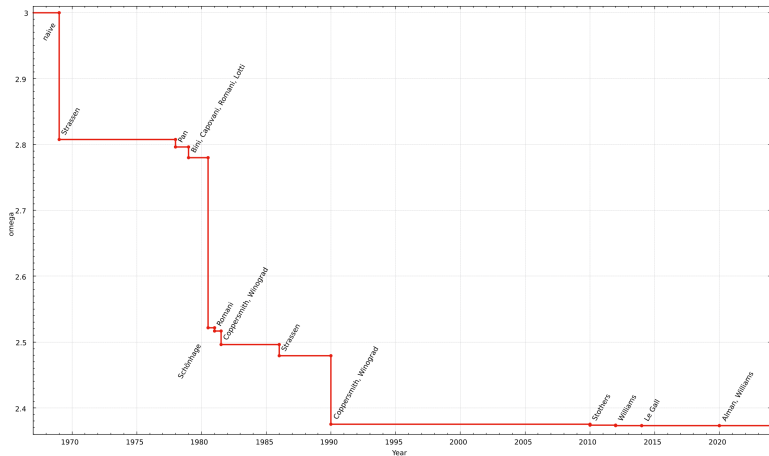
"The optimal number of operations needed to multiply two matrices is still unknown. This is a major open question in theoretical computer science."
(Wikipedia)

Structures de données : performance

Durée (physique) du calcul

Optimisation

Après Éléments d'algorithmique 1-2 (EA3) et Programmation efficace (PE6)



Attention aux Algorithmes Galactiques !

Version 5. Boucles parallèles par blocs

Après Principes de fonctionnement des machines binaires (PF1)

(Déambulation)

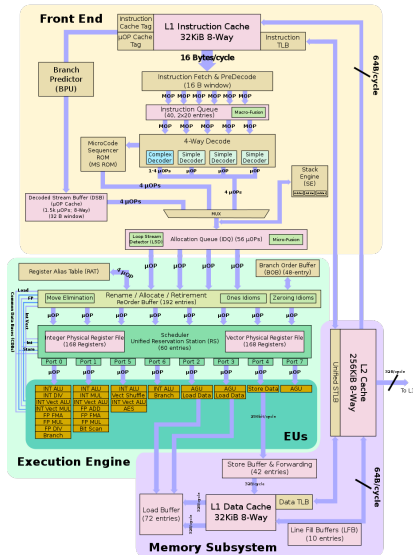
Version 5. Boucles parallèles par blocs

Après Principes de fonctionnement des machines binaires (PF1)

(Déambulation)

Nombre de LoC :	≈ 100
Temps d'exécution :	3.80 s
Accélération relative :	$\times 18.4$
Accélération absolue :	$\times 6\,727$
Fraction de perf. crête :	4.33 %

Éléments d'explication



Version 6. Vectorisation

Après *un stage avec moi*

(Gesticulation)

Version 6. Vectorisation

Après *un stage avec moi*

(Gesticulation)

Nombre de LoC :	≈ 430
Temps d'exécution :	0.41 s
Accélération relative :	$\times 9.4$
Accélération absolue :	$\times 62\,806$
Fraction de perf. crête :	40.45 %

Morale

```
#include <mkl_cblas.h>
```

```
cblas_dgemm(CblasRowMajor, CblasNoTrans, CblasNoTrans,  
            n, n, n,  
            1,  
            (const double *)A, n,  
            (const double *)B, n,  
            0,  
            (double *)C, n);
```

Conclusion⁵



-
5. *"Sunset on Mars with an astronaut looking into the distance"*