

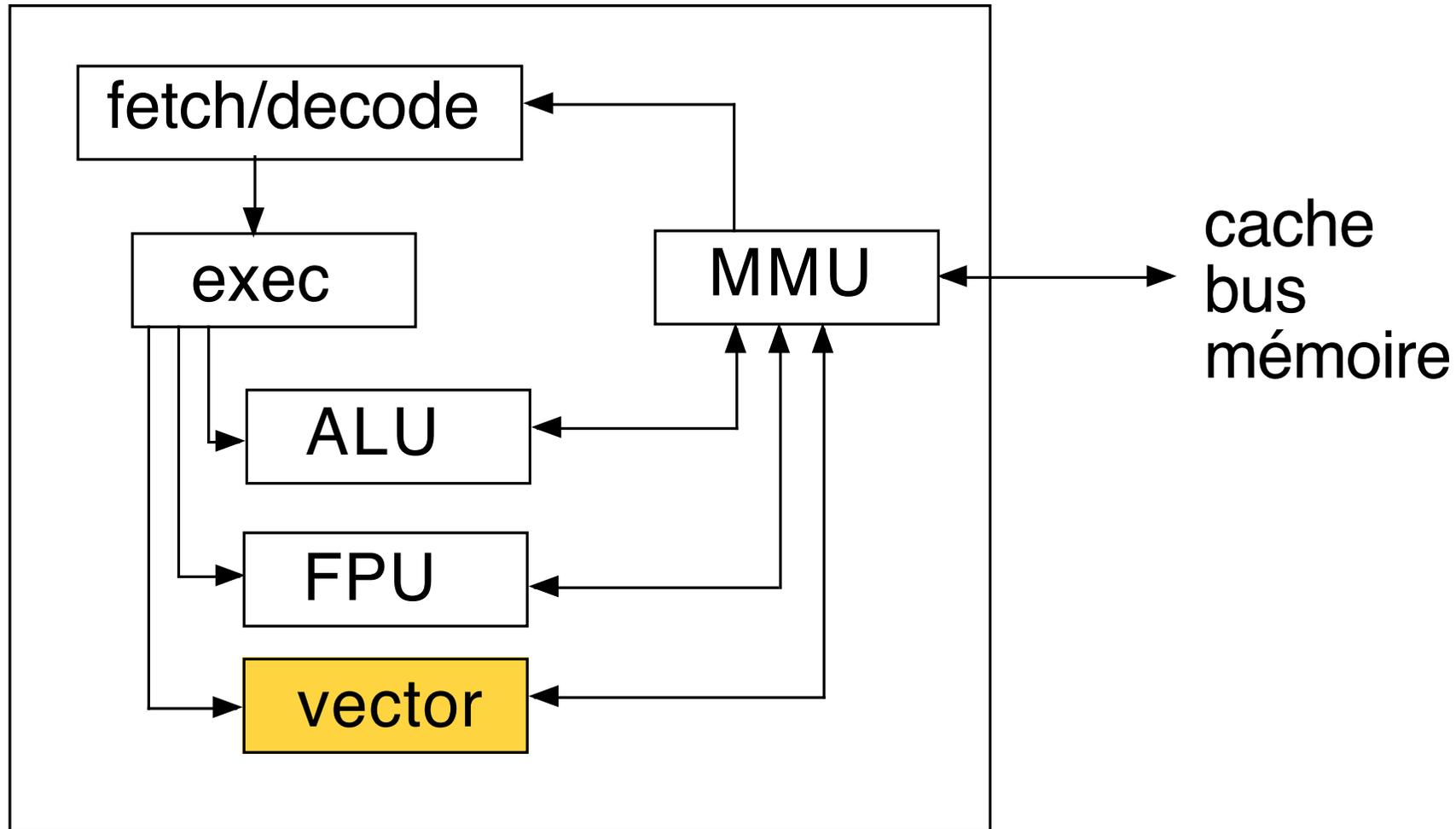
Accélération SIMD d'un calcul matriciel : comparaison de trois plateformes

*Matthijs Douze
IRIT/ENSEEIH*

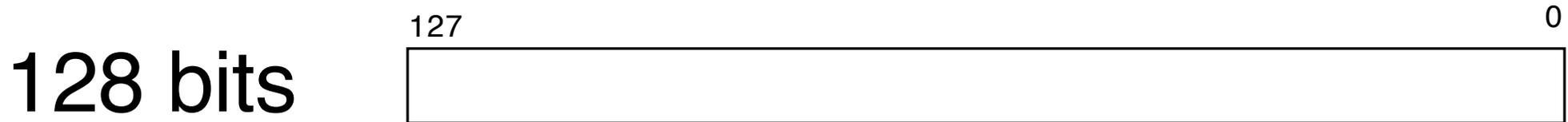
Plan

- Principe
- Utilisation
- Expérimentations

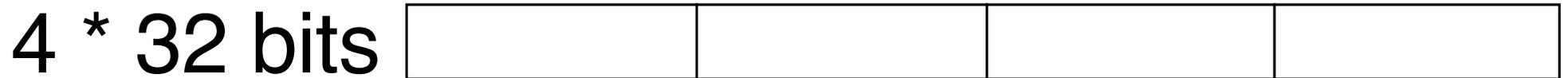
Extension de l'architecture



Registres vectoriels



(signé/non-signé)



(entier/flottant)



Opérations vectorielles

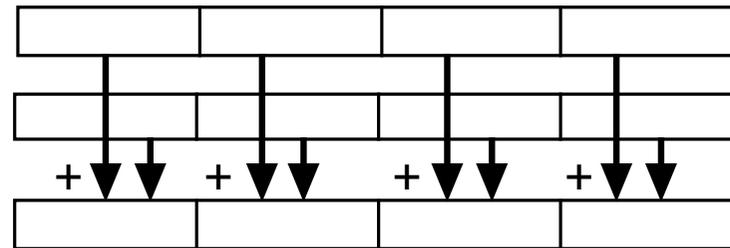
e/s mémoire

ordonnancement

logiques

arithmétiques

comparaisons, min/max



Les 3 implantations

Sun, SPARC, VIS

64 bits, dans FPU, image

PC, IA-32, SSE 2

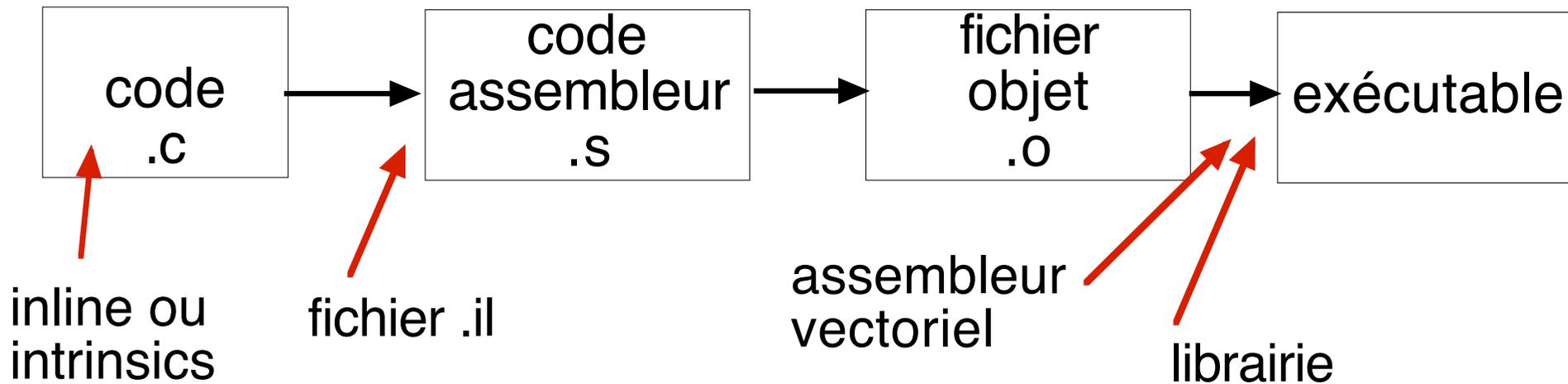
interface avec scalaire

Mac, PowerPC, AltiVec

nombreuses instructions, pas 64 bits

Programmation

depuis C



Librairies

riches

traitement signal & image

convolution, FFT, morphologie, codec

calcul matriciel

BLAS, LAPACK

cryptographie

bignum

Assembleur

défini par le constructeur

fichier à part

appel de fonction depuis C

fichier .il Sun

pas d'appel de fonction

assembleur inline

syntaxe gcc

100% C

intrinsic

un/des types vectoriels
primitives de calcul
syntaxe Motorola

vectorisation automatique
ICC

restrict, aligned

Un problème d'optimisation

résoudre

$$\hat{X} = \arg \min_{X \in \mathbf{R}^{n \times p}} \|AX - B\|$$

où

$$A \in \mathbf{R}^{N \times n}, B \in \mathbf{R}^{N \times p}$$

$$N=5000, n=400, p=8$$

décomposition de Cholesky de $A^T A$
 $n \mid n$ produits scalaires

Mise en œuvre

valeurs entières -255..255

librairies

double/float

pas prévu

vectorisation

instruction spécialisée

par blocs

Résultats

(temps en secondes)

	Mac	PC	Sun
machine	G4 866	P4 1800	US3 Cu 900
scalaire	3.8	1.4	4.1
vectorel	2.1	0.44	2.8
vectorel par blocs	0.54	0.17	1.5

Conclusion

PC rapide, Sun limité

auto-vectorisation de ICC
interface C de Motorola

facile et efficace

gros tableaux d'éléments scalaires

SIMD à la mode