

FastMath.h

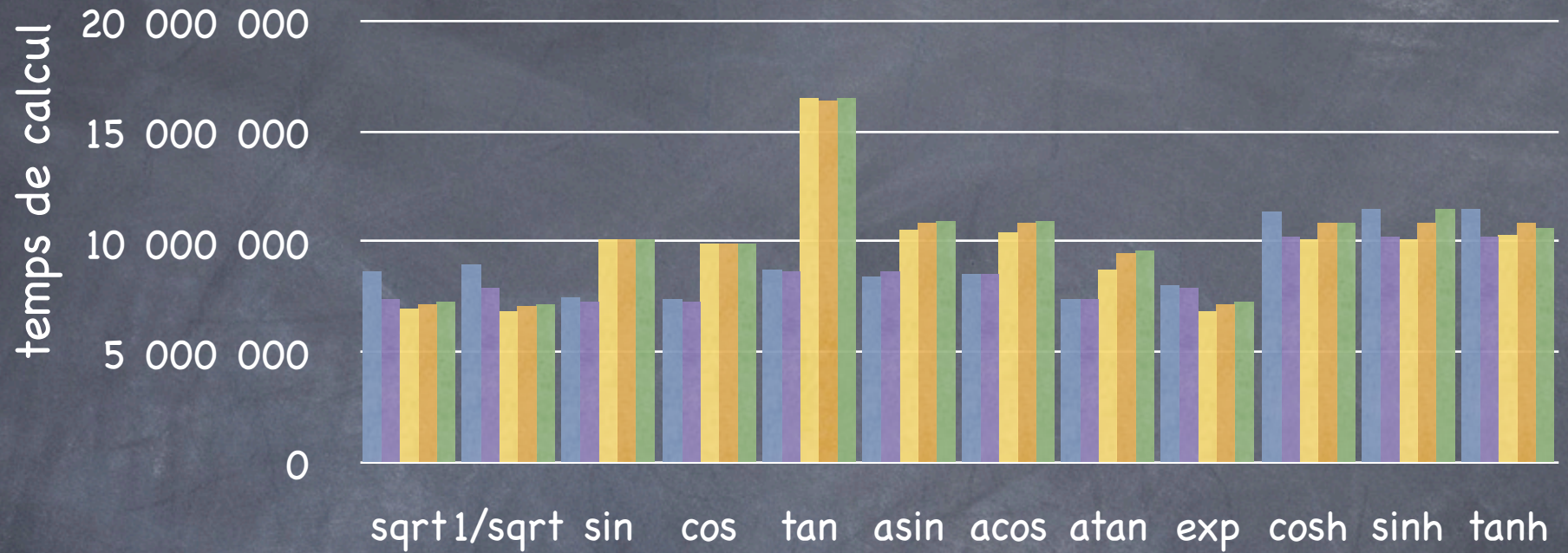
- Tests de fonctions fastSqrt, fastInvSqrt, fastSin, fastCos, fastTan, fastAsin, fastAcos, fastAtan, fastExp, fastCosh, fastSinh, fastTanh
- Selon l'architecture, ces versions fast n'apportent pas forcément un gain (parfois on y perd)
- Ce document présente quelques résultats en fonction des architectures

Significativité

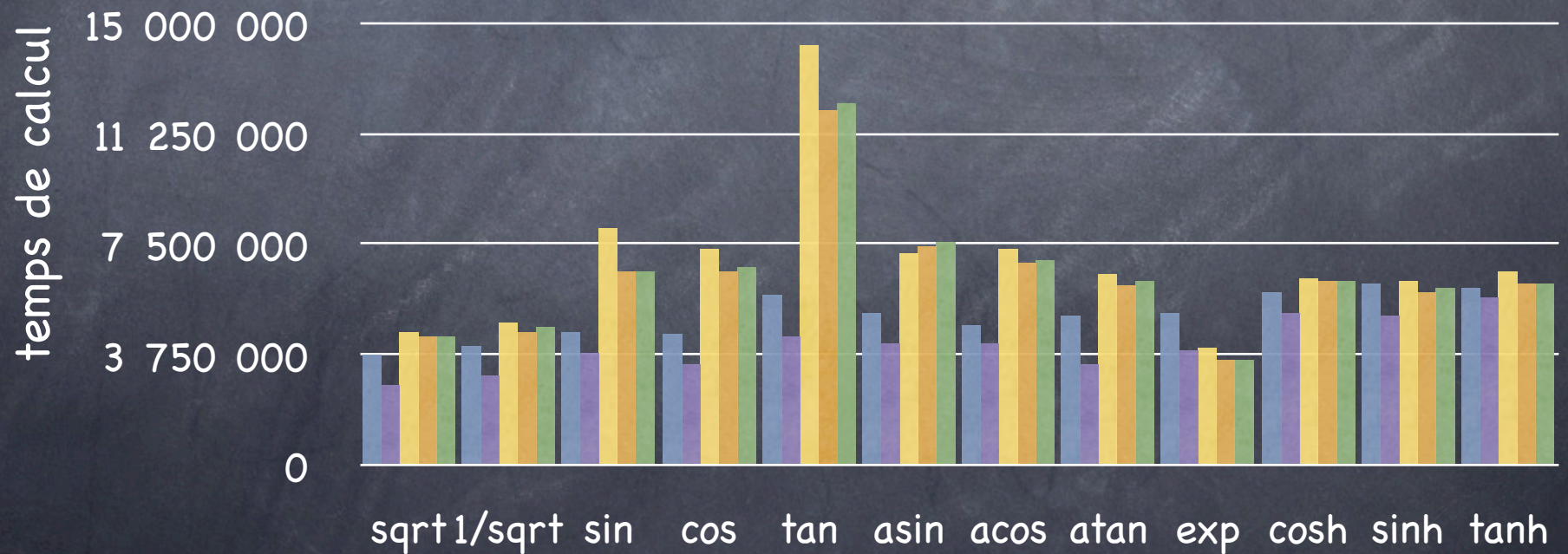
- Tests effectués sur des machines diverses (fréquences, mono/multi pro...) : on ne peut utiliser ces tests pour comparer les architectures entre elles (d'ailleurs les graphiques ne sont pas à la même échelle) ;
- Les données affichées sont les temps de calcul sur un tableau de 20 millions de nombres : une grande barre indique un temps plus long, donc de moins bonnes performances ;
- La valeur absolue importe peu, c'est l'écart relatif des barres, sur une architecture donnée, qui est significatif.

gcc 4.0 -00

G4



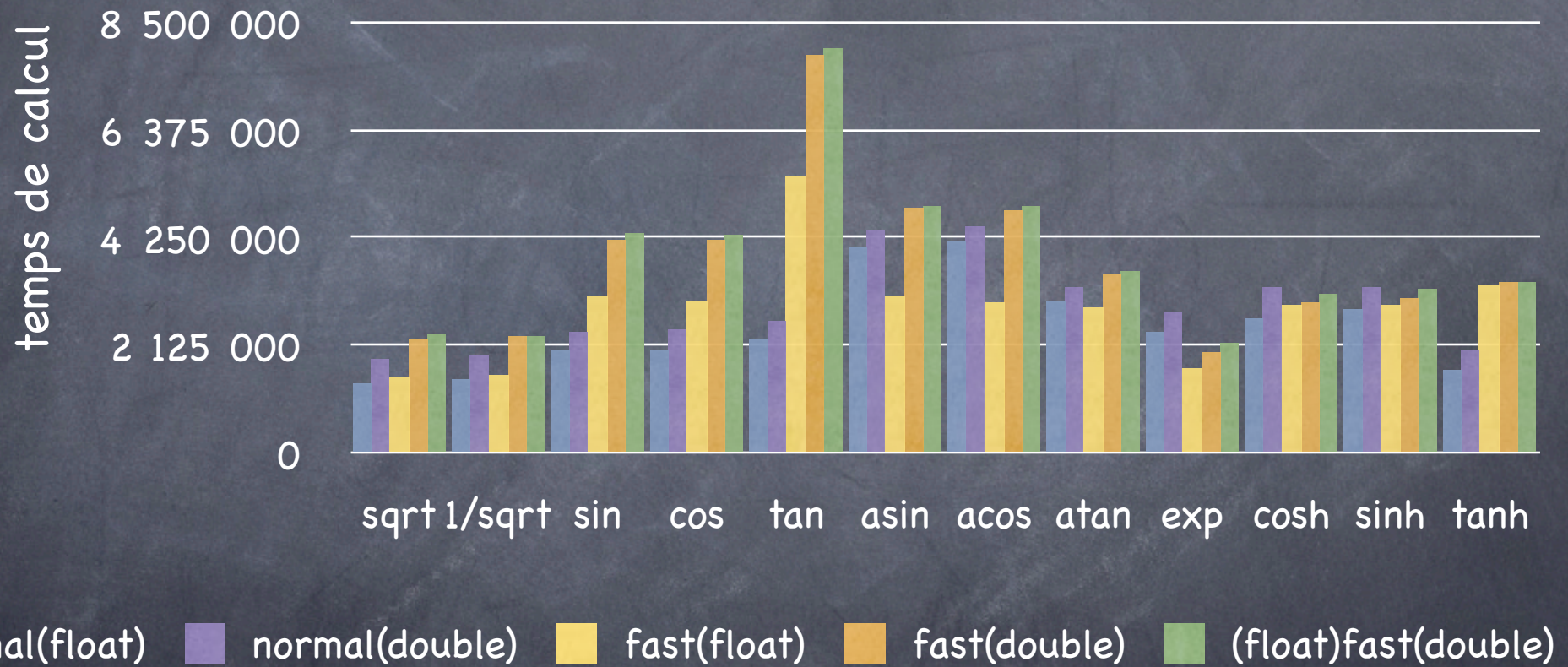
G5



normal(float) normal(double) fast(float) fast(double) (float)fast(double)

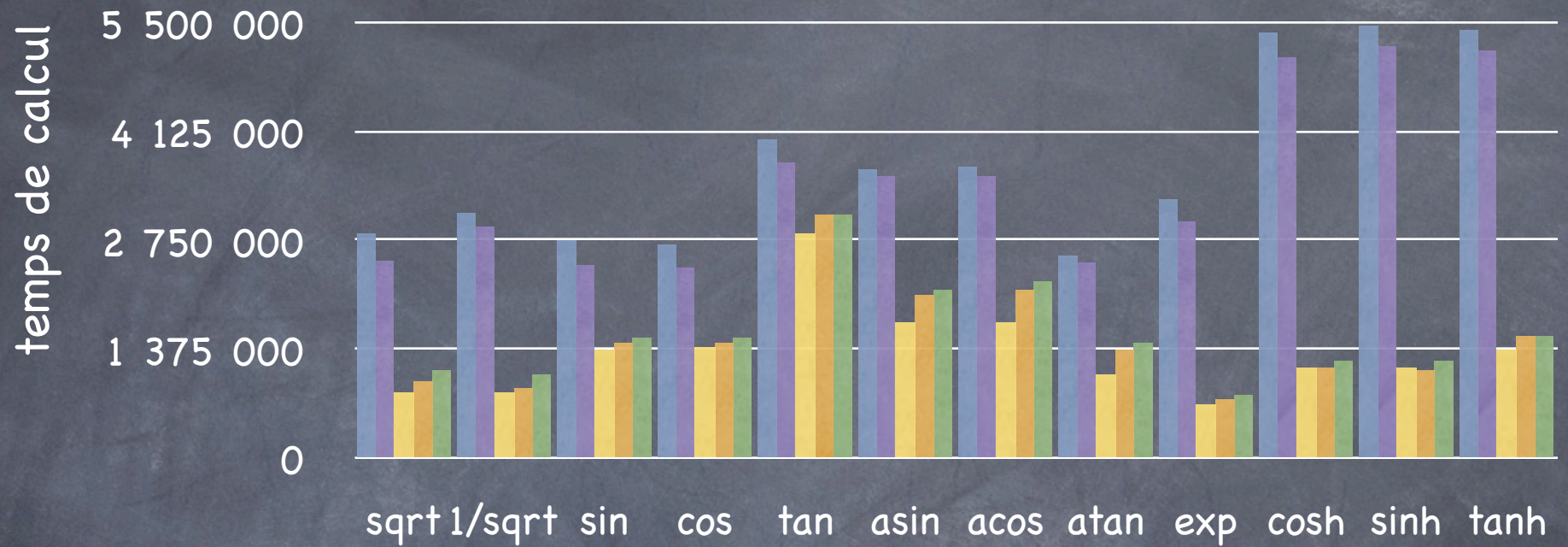
gcc 4.0 -O0

i686

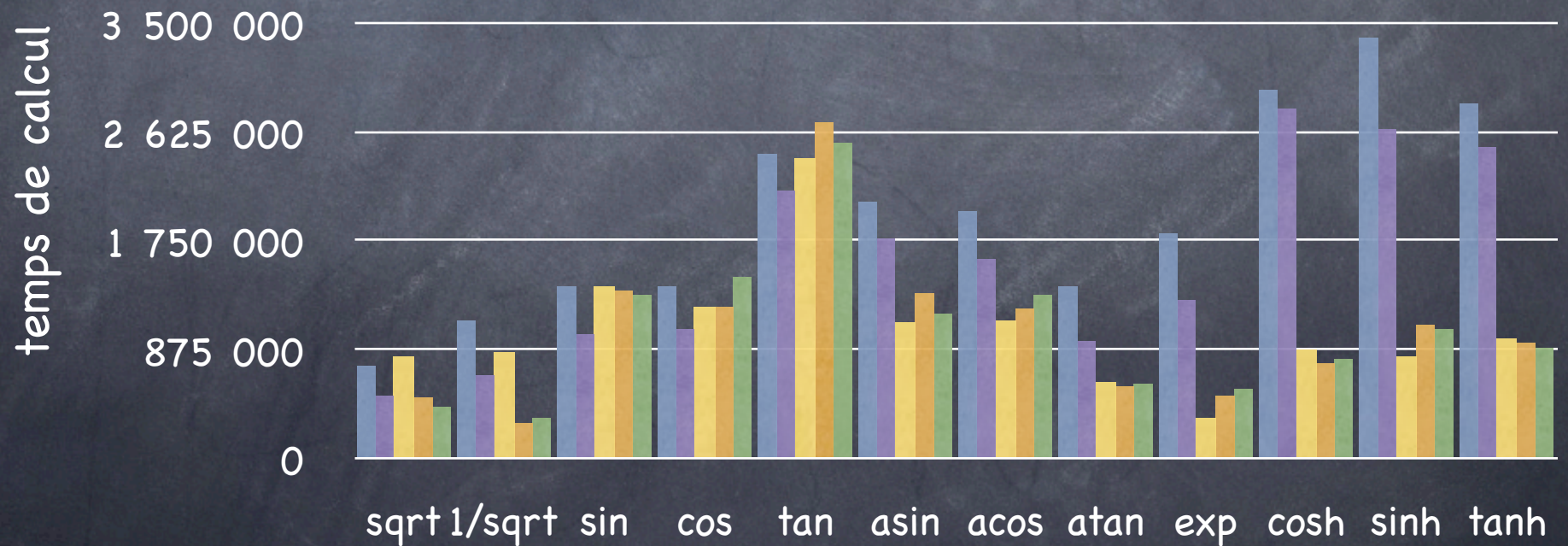


gcc 4.0 -O1

G4



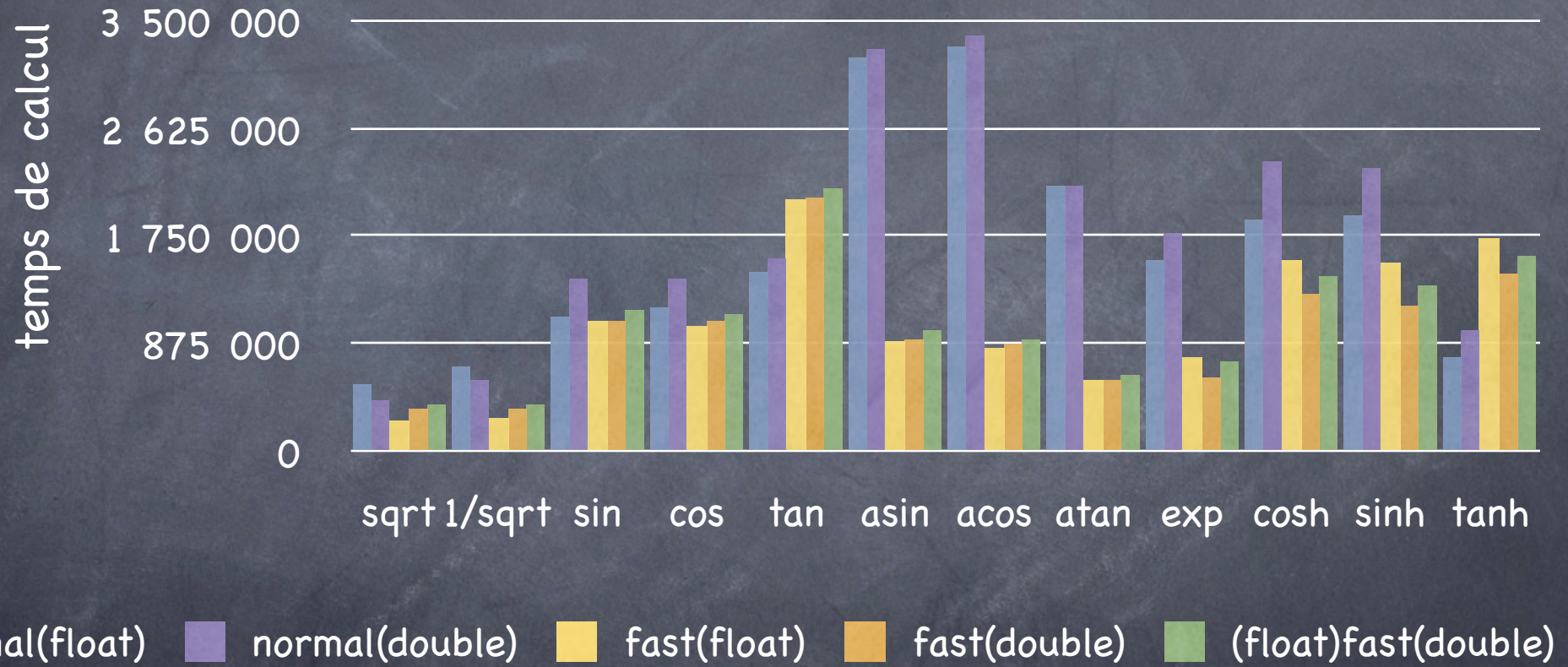
G5



normal(float) normal(double) fast(float) fast(double) (float)fast(double)

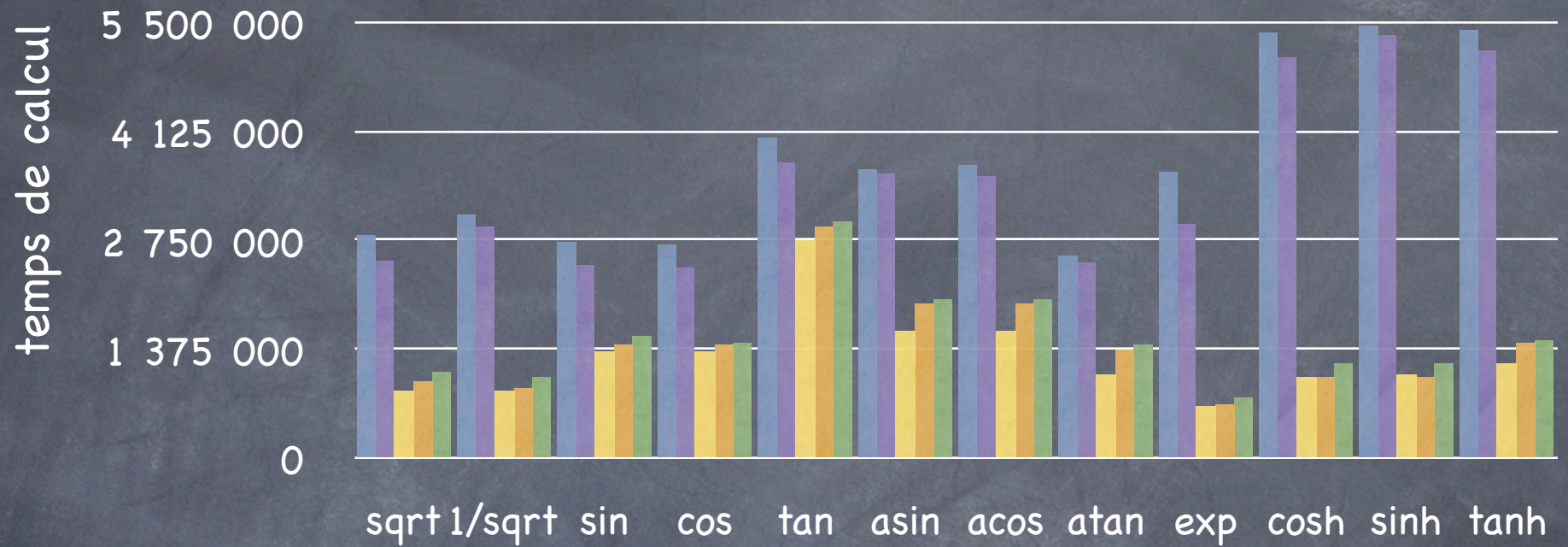
gcc 4.0 -O1

i686

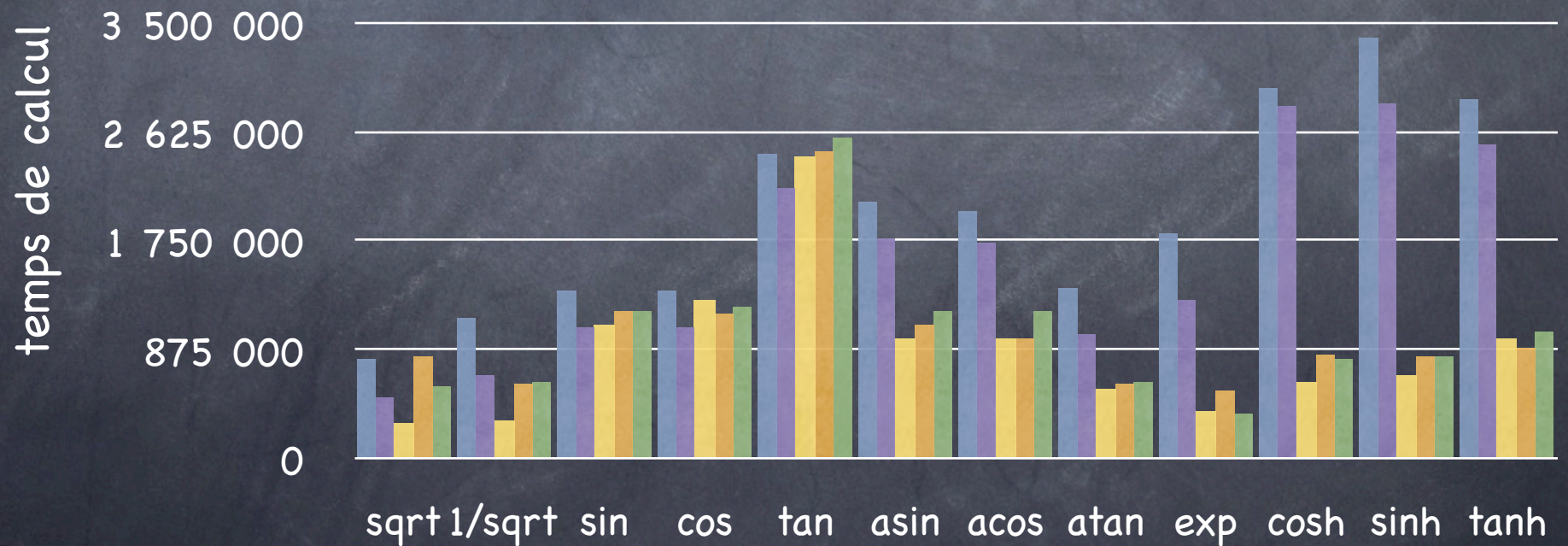


gcc 4.0 -O2

G4



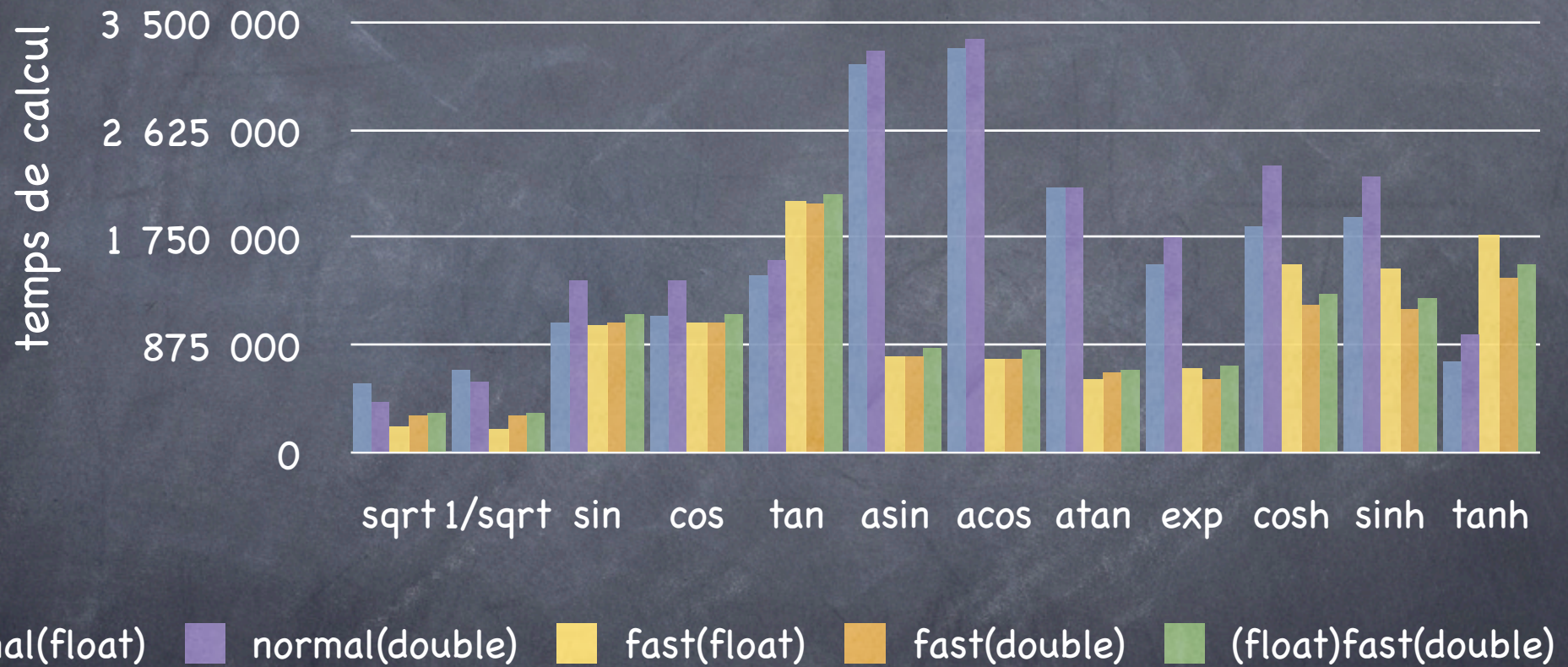
G5



normal(float) normal(double) fast(float) fast(double) (float)fast(double)

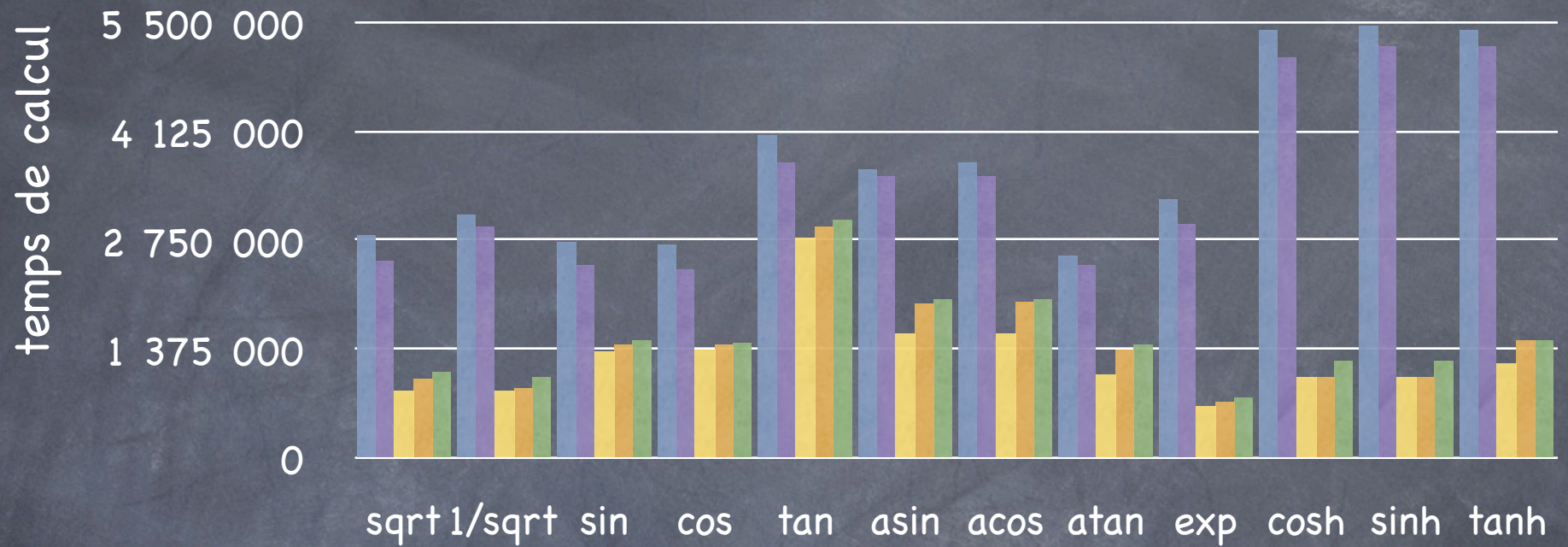
gcc 4.0 -O2

i686

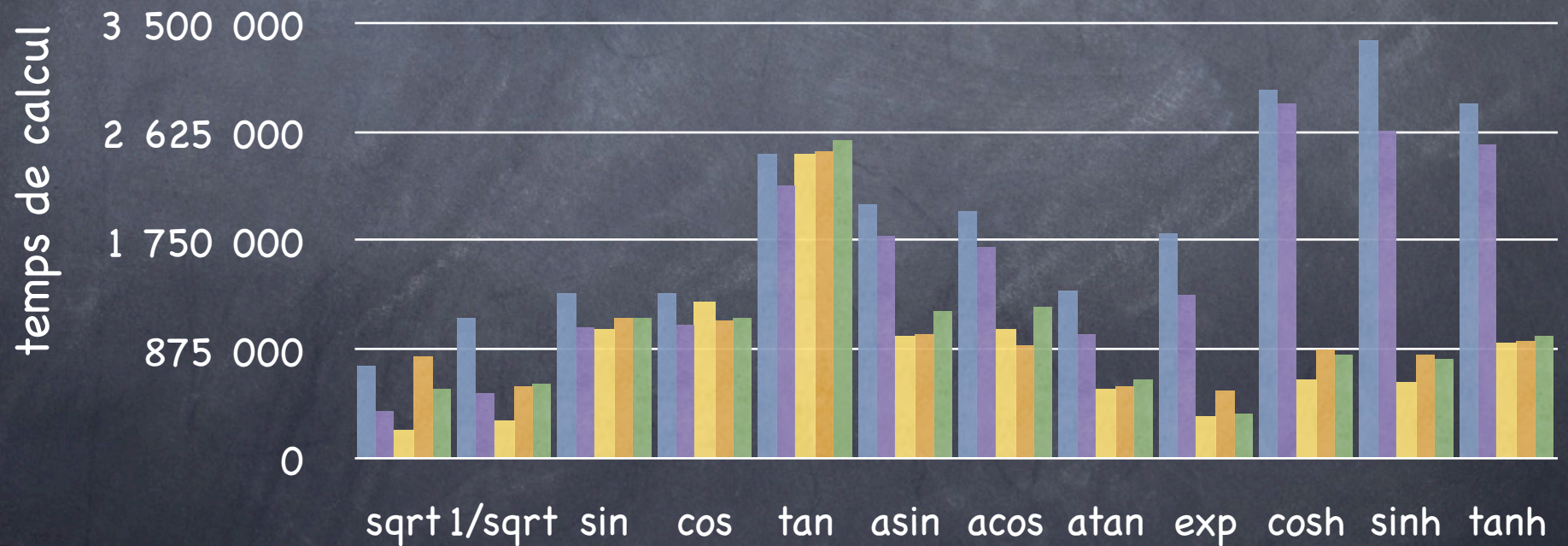


gcc 4.0 -O3

G4



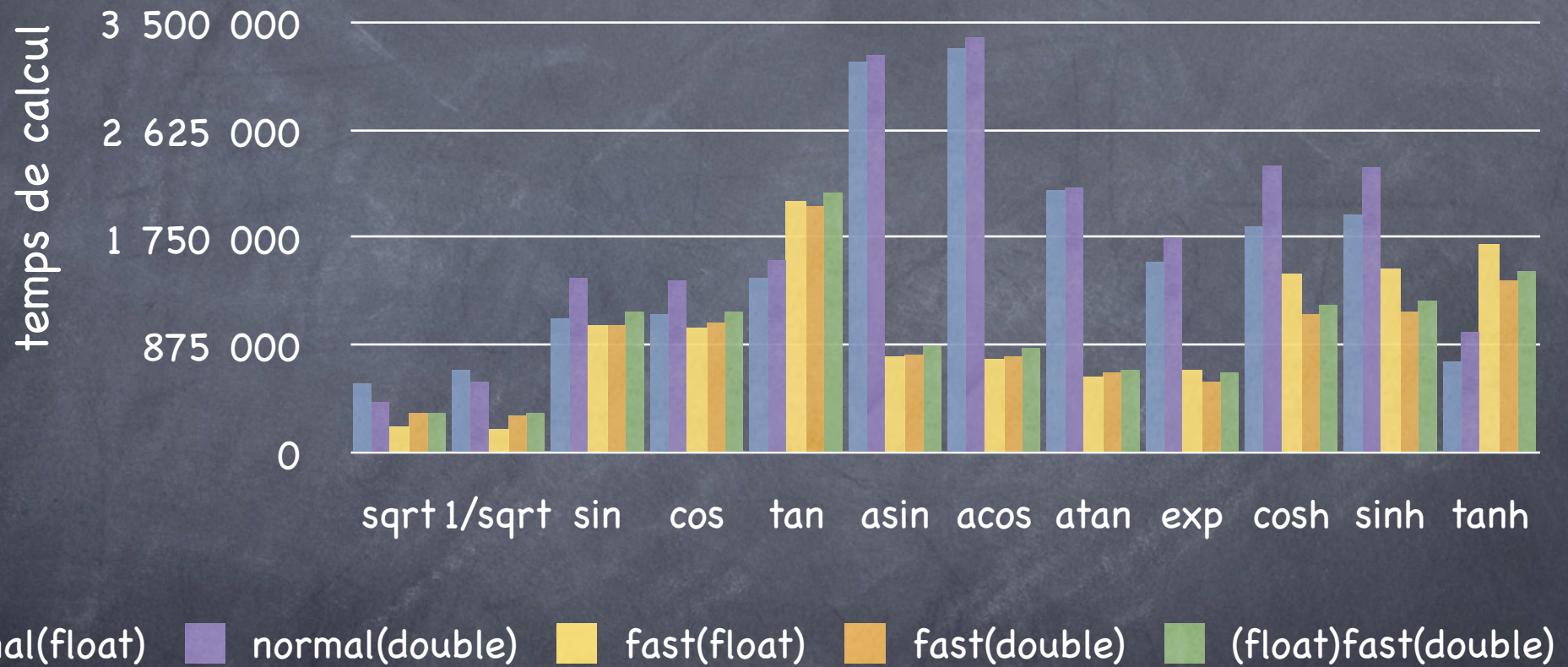
G5



■ normal(float)
 ■ normal(double)
 ■ fast(float)
 ■ fast(double)
 ■ (float)fast(double)

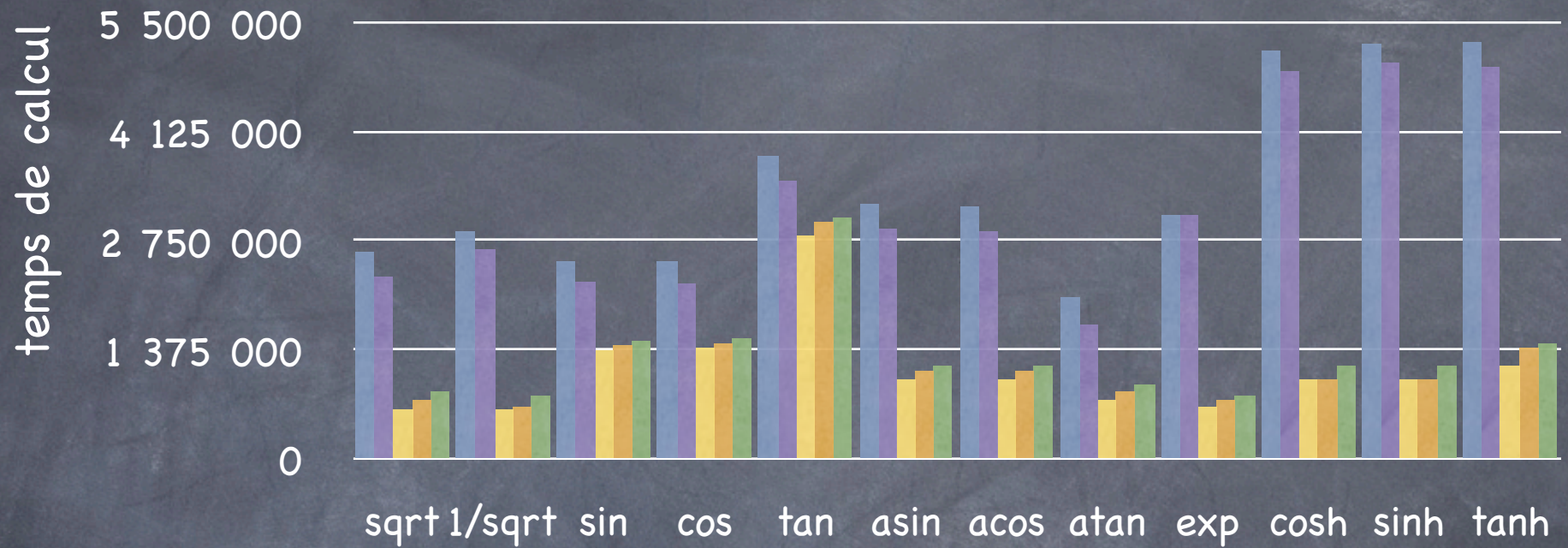
gcc 4.0 -O3

i686

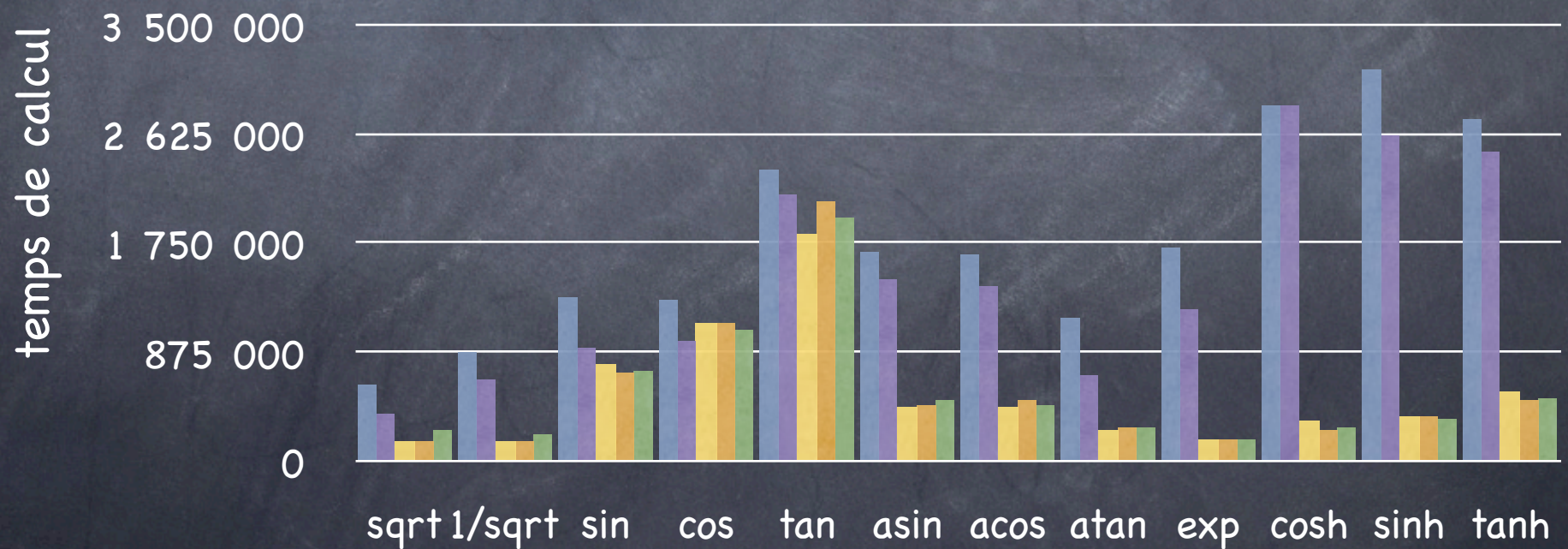


gcc 4.0 -O3 -ffast-math -mtune=... -mcpu=...

G4



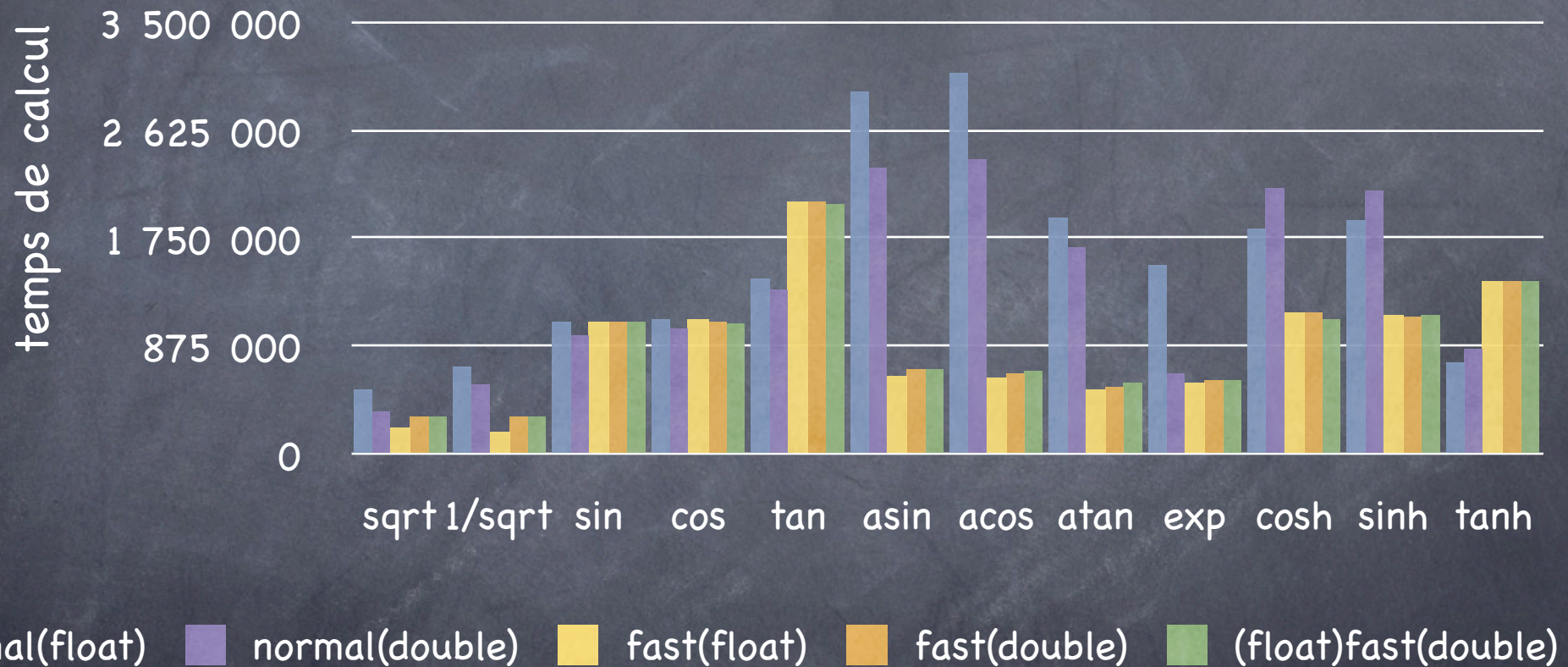
G5



normal(float) normal(double) fast(float) fast(double) (float)fast(double)

gcc 4.0 -O3 -ffast-math -mtune=... -mcpu=...

i686



Observations

- pas de gain, voire perte, en -O0
- gain dès -O1, mais on observe $-O1 \approx -O2 \approx -O3$
- gain supplémentaire en -O3 -ffast-math -mtune=... -mcpu=... uniquement sur G4 et G5
- float n'est pas vraiment plus rapide que double
- Sur G4, toutes les fonctions sont accélérées
- Sur G5, seuls acos, asin, atan, exp, sinh, cosh et tanh sont significativement accélérés; en -O3 -ffast-math, quasiment tout est accéléré
- sur i686, seuls invSqrt, asin, acos, atan, exp, sinh et cosh sont significativement accélérés

Conclusion

- à n'utiliser qu'avec -O1, -O2 ou -O3
- mieux vaut travailler en double, quitte à reconverter en float
- Sur G4, toutes les fonctions sont accélérées
- Sur G5, éviter fastSqrt, fastInvSqrt, fastSin, fastCos et fastTan; si -O3 -ffast-math, fastSqrt et fastInvSqrt sont envisageables
- sur i686, éviter fastSqrt, fastSin, fastCos, fastTan et fastTanh
- On gagne toujours avec fastAsin, fastAcos, fastAtan, fastExp, fastSinh, fastCosh. On gagne aussi avec fastTanh, sauf sur i686