

profiling avec valgrind

Principes généraux de l'optimisation



Premature optimization is the root of all evil
—D.E. Knuth

1. Optimiser l'algorithme, pas le code
2. Ne jamais optimiser un code avec erreurs
3. Ne jamais optimiser sans mesurer

Principes généraux de l'optimisation



Premature optimization is the root of all evil
—D.E. Knuth

1. Optimiser l'algorithme, pas le code (très difficile)
2. Ne jamais optimiser un code avec erreurs (très facile)
3. Ne jamais optimiser sans mesurer (très facile)

Comment trouver des erreurs dans le code

1. **Lecture** : Vérifier que le code est bien formé : indentation correcte, lignes de moins de 80 caractères, fonctions de moins de 25 lignes, max 7 variables par fonction, espace correct autour des signes de ponctuation, etc.
2. **Compilation** : Vérifier que le code compile sans warnings avec plusieurs compilateurs : `gcc`, `clang`, `icc`, `suncc`, `tcc`, `lint`
3. **Execution** : Vérifier que le code tourne sans soucis dans un simulateur, par exemple : `valgrind`, `electric fence`, `totalview`

Une bonne indentation est essentielle



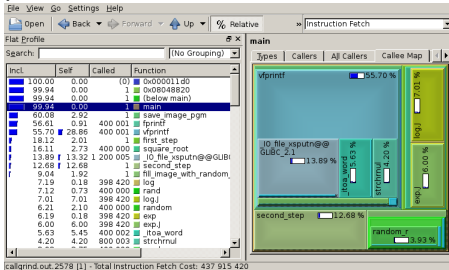
code mal indenté



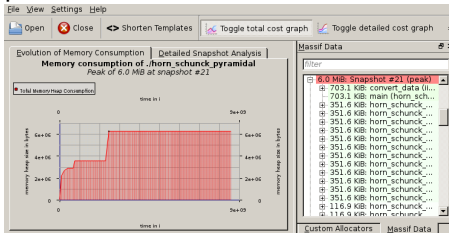
code bien indenté

Comment mesurer la performance d'un code

- Usage de **CPU** : `valgrind --tool=callgrind`
puis visualiser les résultats avec `kcachegrind`



- Usage de **RAM** : `valgrind --tool=massif`
puis visualiser les résultats avec `massif-visualizer`



Comment mesurer la performance d'un code

(exemple interactif)

Résumé

Analyse du code :

`lint`

`clang --analyze`

`gcc -g -Wall -Wextra -Werror -pedantic-errors`

`icc -g -Wall -w2 -Wcheck`

`suncc -g -Xc -errwarn=%all`

`tcc -g -Wall -Werror`

Analyse de l'exécution :

`valgrind --tool=memcheck --leak-check=yes`

`valgrind --tool=callgrind`

`valgrind --tool=massif`

Visualisation :

`kcachegrind`

`massif-visualizer`

OpenMP *profiling*

OpenMP *profiling*

1. Rappel d'OpenMP
2. *Profiling low-tech* (`htop`, `printf`, `clock`)
3. *Profiling high-tech* (`Intel Parallel Studio`)

Intel Parallel Studio

Pièces de Intel Parallel Studio

1. Composer (compilateur)
2. Inspector (contrôle d'erreurs d'exécution)
3. Advisor (?)
4. Amplifier (profilage)