

Comment réaliser des mesures de temps de programmes?

Introduction (emprunté à Valérie Ménissier-Morain)

Pour mesurer le temps d'exécution d'une partie d'un programme, on mesure le temps d'exécution du programme avant l'opération, puis après et par soustraction, on en déduit la durée de l'opération.

Si cette durée est trop faible, elle peut être entachée d'une grosse imprécision. Pour éviter ce phénomène, on exécute l'opération un grand nombre de fois, que l'on nomme *iter*: on effectue la mesure du temps pour les *iter* fois et on divise ensuite le temps total par *iter*.

Cette répétition de l'opération dont on mesure le temps d'exécution peut nécessiter la remise à l'état initial de l'environnement en cas de travail par effet de bord par exemple. Le temps de cette remise à l'état initial est estimé tout à fait négligeable par rapport au temps d'exécution de l'opération et inférieur à l'imprécision due à la mesure du temps pour une seule opération, mais il est utile de s'en assurer rapidement avant de lancer les tests en vraie grandeur.

Un temps d'exécution raisonnablement mesurable est au moins de l'ordre de 2 ou 3 secondes. Il peut évidemment être beaucoup plus important!

Mesure du temps d'exécution

Il est important de mesurer le temps CPU, c'est-à-dire le temps qu'un programme consomme sur le processeur, et non le temps d'exécution pour l'utilisateur. Les commandes `time` des systèmes linux indiquent le temps d'exécution.

En langage C, on peut pour cela utiliser la commande `clock` de la façon suivante:

```
#include <ctime>

int main(){
    clock_t
        temps_initial, /* Temps initial en micro-secondes */
        temps_final;   /* Temps final en micro-secondes */
    float
        temps_cpu;     /* Temps total en secondes */

    temps_initial = clock ();

    ... Code à mesure ...

    temps_final = clock ();
    temps_cpu = (temps_final - temps_initial) * 1e-6;

    return 0;
}
```

Visualisation des temps d'exécution par utilisation de graphiques

Pour cela, vous pouvez utiliser par exemple `gnuplot` ou `xgraphic` ou n'importe quel logiciel d'édition de courbes.

Analyse et comparaison des courbes obtenues

La partie essentielle du travail n'est pas de dessiner des courbes mais leur **analyse**. On attend là à la fois:

- une analyse de la croissance de chaque courbe, mettant en évidence la complexité de chaque implémentation constatée sur le jeu d'exemples choisi
- une comparaison des courbes obtenues, pas seulement globale, mais aussi 2 à 2. Par exemple, si on compare trois courbes, on s'attend non seulement à un commentaire du style « telle courbe est toujours en dessous des deux autres », mais aussi à une comparaison des deux autres courbes; une explication de la plus grande efficacité de cette méthode par rapport aux deux autres,...
- mais aussi l'analyse de la pertinence et des limites du jeu de tests et la comparaison des résultats constatés avec les estimations théoriques de complexité.