Présentation Perf

Jérémy ZANGLA

April 5, 2020

Contents

1 Présentation

1.1 Profiling

Le *profiling* est le processus qui analyse l'exécution d'un programme. Il permet d'étudier les performances de notre application, afin de trouver les points à améliorer. Les domaines que l'on peut étudier sont diverses : temps d'éxécution, espace mémoire, nombre et fréquence d'appel des fonctions. On appelle *profiler* l'outil qui nous permet de faire du profiling.

1.2 Outils

Il existe différents outils, appelés *profilers*, permettant le profiling en fonction du langage de programmation choisis. On citera par exemple :

- C/C++ : Valgrind (Callgrind, Cachegrind, ...), ...
- Java : Java Runtime Analysis Toolkit (JRAT), ...
- Python : cProfile, ...

Certains de ces outils doivent être intégrés au programme, d'autres sont lancés indépendamment en donnant le programme en paramêtre (les sources dans le cas d'un langage interprété).

2 Perf

2.1 Présentation

Le profiler traité ici est perf, il permet d'analyser des programmes développés dans plusieurs langages. Il ne fonctionne cependant que dans un environnement linux. Nous allons ici nous concentrer sur son utilisation sur des programmes en C++. C'est un programme qui crée un sous-système linux, permettant ainsi d'avoir son propre noyau pour acquérir les données d'exécution du programme analysé. Il est maintenu en parallèle du noyau linux et reçoit donc des mises à jour régulières. Il est séparé en plusieurs modules, chacun possédant des fonctions particulières. Cet utilitaire peut faire bien plus que ce qui sera présenté ici.

2.2 Installation

Ce logiciel n'est disponnible que sous Linux. La plupart des distributions permettent de l'installer directement depuis les dépots de leur gestionnaire de paquets (sous ubuntu on installera *linux-tools-common*). Cependant on peut l'installer manuellement depuis un linux, en clonant le dépôt git suivant :

git://git.kernel.org/pub/scm/linux/kernel/git/stable/linux-stable.git
Il faut ensuite passer sur la branche correspondant au noyau du système utilisé. On peut ensuite se rendre dans le sous-répertoire *tools/perf*, et excécuter *make*. Si une erreur est levée, il faut vérifier que les programmes suivant sont installés : *make*, *flex* et *bisons*.

2.3 Configuration

2.3.1 Droits d'utilisation

Pour pouvoir excécuter perf, il faut par défaut avoir des droits super-utilisateurs. Il est cependant possible de changer ce comportement en changeant le contenu du fichier de configuration : */proc/sys/kernel/perf_event_paranoid*. Il faudra également modifié le fichier */etc/sysctl.conf* pour que les changements soient permanents. Le niveau par défaut est 3. Plus le niveau

est faible, plus vous aurez accès à des parties du programme sans avoir besoin de droits supplémentaires (avec un minimum à -1 où l'accès est total). Sur une installation neuve, l'utilisation de *perf stat -a* vous indiquera les différents niveaux de privilèges possibles. La commande *perf test* permet de savoir quels évènements peuvent être analysés par perf dans la configuration actuelle.

2.3.2 Compilation

Vous pouvez utilisez perf sur tous les programmes. Cependant pour pouvoir avoir un lien avec le code source il faudra compilé du C++ avec les options -g et -pg de g++. Sans ces options les noms de fonctions seront remplacés par leur adresse mémoire (sous forme hexadécimale).

2.4 Utilisation

2.4.1 Affichage des statistiques systèmes

```
1 perf stat <executable>
```

Cette commande exécutera alors le programme. Puis à la fin, il affichera quelques statistiques sur les resources utilisées. Les informations affichées dépenderont alors des resources utilisées, par défaut le programme affichera ce qu'il juge de plus pertinent. On aura donc par exemple le temps d'exécution du programme, le nombre de changement de contexte ou bien le nombre d'instructions processurs.

L'utilisation du paramêtre *-d*, permet de rajouter du détail dans les informations affichées notemment l'utilisation du cache du processeur. Il est possible de spécifier plus précisémment ce que l'on veut analyser, la page du manuel concernant la commande regorge d'informations.

2.4.2 Analyse graphique

L'analyse d'un programme se fait grâce à l'enregistrement d'un fichier contenant l'ensemble des informations.

```
1 perf timechart record <executable>
```

Cette commande génère un fichier, qui sera utilsé pour afficher des informations sur l'éxécution du programme. On peut ensuite générer une image vectorielle (au format svg) qui contiendra un graphique d'exécution du programme.

```
1 perf timechart
```

L'image peut ensuite être ouverte dans n'importe quel programme gérant les svg, chromium par exemple peut le faire.

2.4.3 Analyse des appels

Un second fichier peut être généré par perf pour analyser un programme. Il faut ensuite utiliser une autre partie de perf pour étudier ce fichier.

```
1 perf record
2 perf report
```

La seconde commande nous affichera alors sur quelles fonctions le processeurs a passé le plus de temps. L'interface que nous donnes le programme permet la navigation dans ces différents appels. On notera la présence d'un plus sur la première colonne du terminal indiquant si la ligne peut être validé, avec la touche entrée, pour avoir plus de détails. L'option *–stdio* permettra d'afficher en texte uniquement le résultat de la commande, au lieu d'avoir une interface.

En ajoutant l'option *-g* lors de l'utilisation de *perf record* et avec *-n -g folded* sur la seconde commande, on obtient une liste des piles d'appels avec leur fréquence d'apparition.

2.5 Extensions

Il existe des outils tels que FlameGraph qui permettent d'utiliser les fichiers de données brutes enregistrés par perf pour analyser différemment ces données. Une description de cette outils est disponnible la page http://www.brendangregg.com/FlameGraphs/cpuflamegraphs.html

3 Sources

Perf est une suite logiciel très complète qui permet d'avoir une analyse très précise d'un système ou d'un de ses programmes. L'utilisation du manuel est très importante pour comprendre son fonctionnement plus en profondeur.

On peut également chercher sur internet. Les pages suivantes sont très utiles pour comprendre plus rapidement comment agencer les options de ce logiciel pour arriver à faire ce que l'on souhaite :

- http://www.brendangregg.com/perf.html
- https://www.fosslinux.com/7069/installing-and-using-perf-in-ubuntu-and-centos.htm