Projet de recherche de L3

Comparaison de fonctions de hachage

Florian REYNIER & Mathis CARISTAN

30/01/2016

Résumé

L'objectif du projet est de comparer différentes fonctions de hachage (non-cryptographique) du point de vue de leur complexité algorithmique et de leur uniformité dans la répartition des clés. On étudiera plusieurs fonctions utilisées couramment en informatique, que l'on testera sur différents fichiers source, de taille et de contenus variables.

1 Le hachage

Le hachage est un procédé qui utilise une fonction de hachage pour associer à un paramètre en entrée, une empreinte unique. Ceci permet plusieurs applications, notamment un accès rapide à des grandes bases de données, ou d'établir certaines relations (l'égalité ou la non-égalité) entre des données sans accéder directement à celles-ci. Le hachage permet également de ramener un grand espace de données (potentiellement infini), à un espace d'empreintes fini et de taille connue.

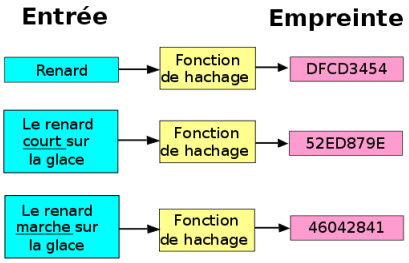

FIGURE 1: Illustration d'une fonction de hachage.

Il existe également des fonctions de hachage cryptographique à "sens unique" qui assurent une reconstruction des données à partir de l'empreinte impossible par calcul. Ces fonctions peuvent être utilisées pour vérifier qu'une entrée est associée à une empreinte donnée (dans l'authentification par mot de passe sous UNIX par exemple), mais nous n'étudierons pas ces fonctions dans le cadre de ce projet.

2 Méthodologie

2.1 Les outils

Pour ce projet, nous allons utiliser l'OCaml comme langage de programmation. `(TODO : POURQUOI)` De plus, nous utiliserons GitHub pour la gestion du code et des ressources [1].

2.2 Les fonctions

Sélection des fonctions (raisons ?)

2.3 Les clefs

Nous comptons constituer des corpus de différents types de clefs. Les types de clefs que nous envisageons pour le moment sont :

- Du texte en français,
- Du texte en anglais,
- Du code,
- Des listes de log,
- Des clés aléatoires,
- Des QRcodes,

Nous constitueront les corpus à partir de sources libres de ces types de données, notamment la base de donnée du projet Gutemberg et des projets sur GitHub.

2.4 Points de comparaison

L'uniformité de répartition des clefs (et le nombre de collision créées), ainsi que les vitesses d'exécution des algorithmes basés sur les diverses fonctions seront les points de comparaison principaux sur lesquels nous comptons travailler.

1. https ://github.com/Kiskuit/L3Project.git

3 Analyse

Pour évaluer la distribution des empreintes sur la taille de l'espace des empreintes, nous allons nous baser sur une analyse statistique des résultats. Dans le cas où une fonction viendrait à s'éloigner d'une distribution uniforme, il sera également possible de réaliser des ajustements de la distribution par différentes fonctions, et de chercher l'origine de cette irrégularité.

4 (Discussion)

(Si besoin et idées)