

TP C++ n° 4 :

Analyse de logs apache

DURAND Bastien

FOURNIER Romain

6 février 2018

Table des matières

1	Cahier des charges	2
2	Conception	2
3	Données	3
4	Réalisation	4

Introduction

1 Cahier des charges et spécifications

Nous avons créé un programme d'exploitation de fichiers log qui respecte les besoins formulés dans l'énoncé : -par défaut, l'outil affiche dans la console sous forme textuelle la liste des 10 documents les plus populaires par ordre décroissants. L'outil s'appelle en ligne de commande par la commande : `./analog [options] nomfichier.log`. -l'outil peut prendre 3 options en ligne de commande :

- `[-g nomfichier.dot]` produit un fichier au format Graphviz.
- `[-e]` permet d'exclure les ressources qui ont une extension js, image ou css.
- `[-t]` permet de ne prendre en compte que les hits dans un créneau horaire correspondant à l'intervalle [heure, heure+1[.

L'expression initiale des besoins étant un peu floue, nous y avons ajouté quelques spécifications :

- Par défaut, l'outil affiche dans la console sous forme textuelle la liste des 10 documents les plus populaires par ordre décroissants.
 - Si il y a moins de 10 documents consultés, l'outil donnera la liste de tous les documents classés par popularité.
 - Si la recherche n'a retourné aucun documents, l'outil donnera une réponse prédéfinie.
 - Si le fichier .log est vide, inexistant, mal formé ou non spécifié, l'outil affiche l'aide dans le terminal.
- `[-g nomfichier.dot]` produit un fichier au format Graphviz.
 - Pour les ressources n'ayant pas de referer (accès direct à la ressource), un noeud " - " représente les accès directs dans le graphe.
 - Si le nom du fichier .dot n'est pas spécifié, l'outil affiche l'aide dans le terminal.
 - Si un fichier du même nom existe déjà, une confirmation est demandée à l'utilisateur avant d'écraser le fichier.
- `[-e]` permet d'exclure les ressources qui ont une extension js, image ou css.
 - Si une ressource n'a pas d'extension, elle sera sélectionnée.
- `[-t]` permet de ne prendre en compte que les hits dans un créneau horaire correspondant à l'intervalle [heure, heure+1[.
 - L'outil devra aussi fonctionner si la plage horaire s'étend sur 2 jours.
 - Si l'heure n'est pas spécifiée ou au mauvais format, l'outil affiche l'aide dans le terminal.

2 Conception

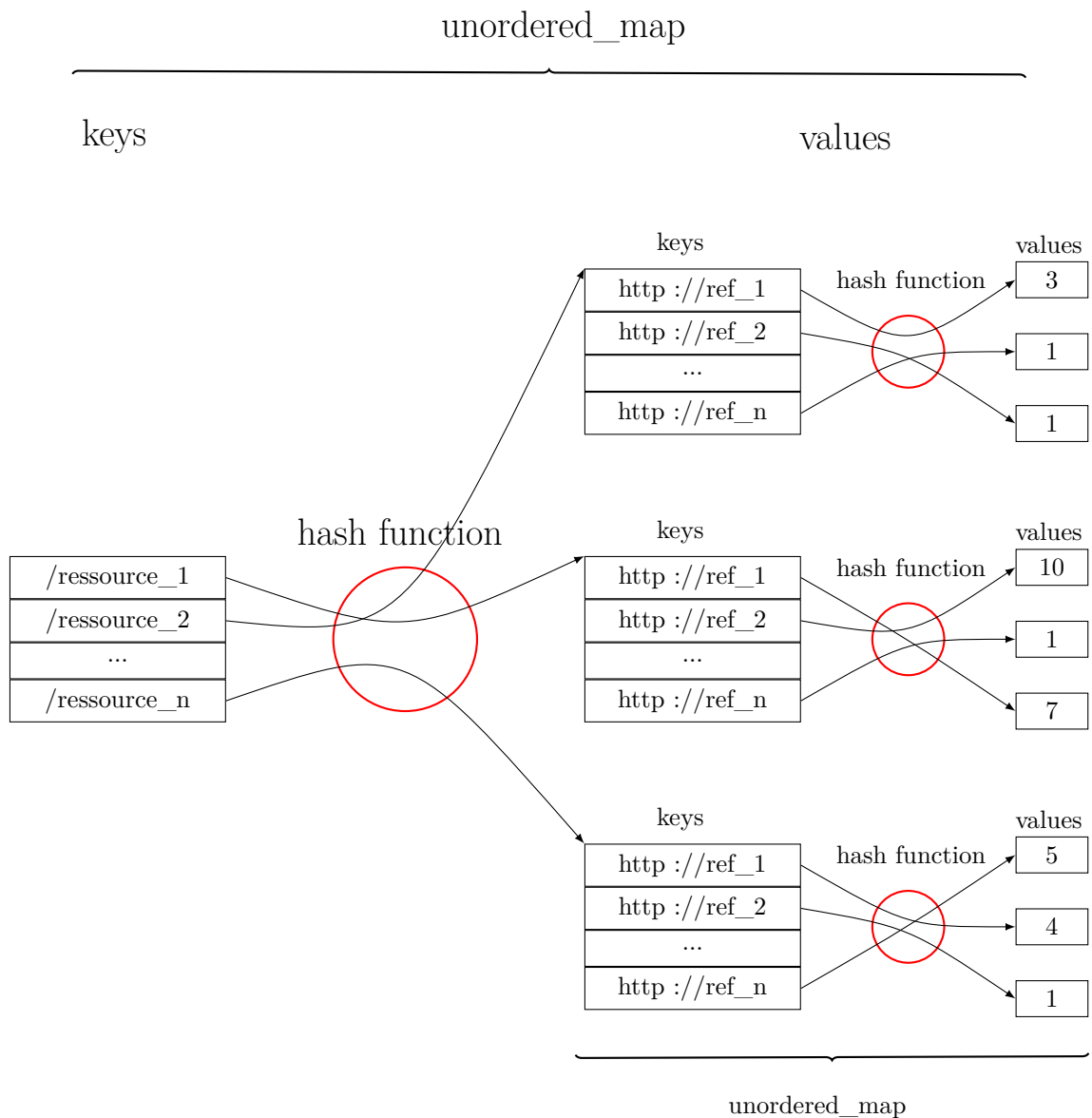
Notre application ne comporte qu'une seule interface et une seule réalisation associée. En effet, nous avons décidé d'utiliser une structure de données de la STL pour gérer nos données. Nous n'avons donc pas eu à implémenter de classes supplémentaires.

3 Données

Les données tirées de l'exploitation des logs seront stockées dans plusieurs conteneurs de la stl de type "unordered_map". A la différence d'une "map" ce conteneur ne stocke pas ses clés triées. Ainsi ce n'est pas sous forme d'arbre binaire rouge et noir qu'une "unordered_map" est implémentée mais sous la forme d'une table de hachage. Le cout en insertion ou recherche est en $O(1)$ en moyenne et $O(n)$ dans le pire des cas.

Les données extraites des logs sont des ressources et des referers. Voici la manière dont nous les stockons :

- chaque ressource est une clé dans une "unordered_map" "générale",
- clé étant associée à une valeur de type "unordered_map",
- les referers pointant sur une ressource sont clés de l'"unordered_map" liée à cette ressource,
- clé étant associée à une valeur de type int,
- cet entier désignant le nombre de fois que la ressource a été demandée via ce referer.



4 Réalisation