Analyse de Log Apache

TP C++3 utilisation de la STL

Sommaire

[Introduction 3](#__RefHeading__442_2025629730)

[Spécifications 3](#__RefHeading__444_2025629730)

[Spécifications détaillées 3](#__RefHeading__235_948620505)

[Architecture 4](#__RefHeading__446_2025629730)

[Classes 4](#__RefHeading__448_514793285)

[Structures de données 5](#__RefHeading__450_514793285)

[Manuel 6](#__RefHeading__327_948620505)

# Introduction

Le web est basé sur un échange de données entre un client (le navigateur) et un serveur. Le client effectue une requête par l'intermédiaire du protocole HTTP (HyperText Transfer Protocol). Cette requête est transmise via le réseau vers le serveur. Le serveur fait en retour une réponse consistant notamnent en un code de retour, et un document. Toutes ces opérations sont consignées par le serveur dans un fichier journal ou *log* en anglais.

On se propose d'extraire de l'information de ces fichiers de logs.

# Spécifications

Le programme est un outil en ligne de commande qui permet d'analyser un fichier de logs Apache. Il devra à chaque exécution afficher les 10 pages les plus visitées et leur nombre de visites. Si l'utilisateur le demande via l'option -g <nomfichier>, il pourra générer un fichier GraphViz représentant les passages d'une page à une autre et le stocker dans le fichier fourni en paramètre.

L'utilisateur pourra exclure des calculs les fichiers d'images, de formatage et de scripts via l'option -e.

Il pourra aussi restreindre les calculs sur une plage horaire de son choix via l'option -t heure, avec heure dans l'intervalle [0,23] (heure GMT, et non locale).

L'utilisateur devra inscrire dans le fichier TP\_STL.cfg l'URL de base du serveur, sans quoi cette dernière ne sera pas prise en compte.

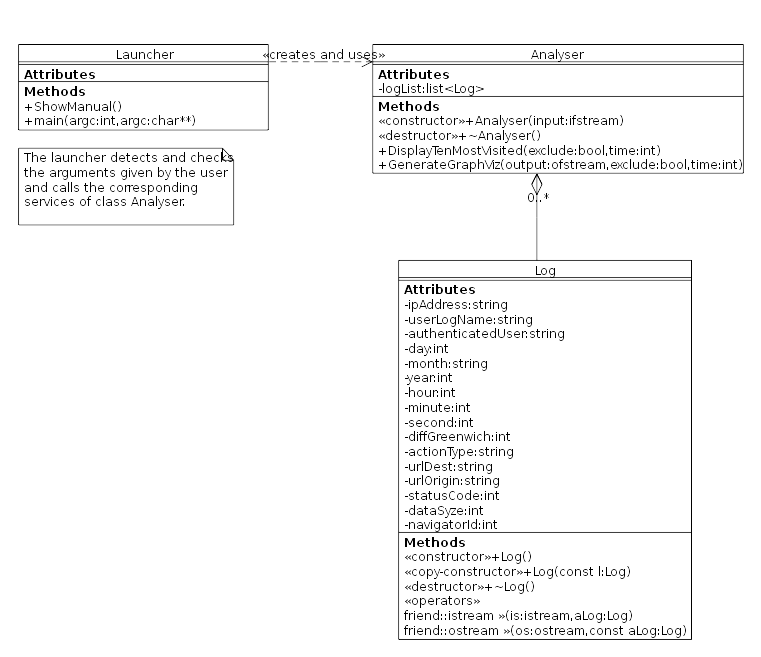
## Spécifications détaillées

Le tableau suivant récapitule les spécifications et leur tests associés. Le numéro correspond au répertoire Tests/Test<numéro> dans lequel est décrit le test selon le framework de test fourni.

| Description | Numéro |
| --- | --- |
| **– Détection des arguments et vérification de leur validité –** | 0.\* |
| L'appel du programme sans aucun paramètre déclenche l'affichage du manuel. | 0.1 |
| L'appel avec un seul paramètre doit correspondre au fichier de logs à analyser. S'il n'est pas présent dans le répertoire, une erreur doit être levée. | 0.2 |
| L'appel avec l'option -e puis le nom du fichier (présent dans le répertoire, mais vide) doit être détectée. Aucune erreur ne doit être levée. | 0.3 |
| L'option -t nécessite un nombre après. Si l'appel place l'argument -t en avant dernière position (avant le fichier de log), le programme doit détecter une erreur. | 0.4 |
| L'option -t nécessite un nombre dans l'intervalle [0,23]. Si le nombre fourni ne rentre pas dans cet intervalle, une erreur est levée. | 0.5 |
| Le programme doit être capable de détecter toutes les options, peu importe l'ordre de celle-ci. L'option -g qui nécessite un nom de fichier après elle pour stocker le graphe lève une erreur (similaire au test 0.4). | 0.6 |
| Lors d'un appel correct avec toutes les options, toutes celles-ci sont détectées et ne lèvent pas d'erreurs. | 0.7 |
| **- Lecture des données -** | **1.\*** |
| Le programme doit être capable de lire un fichiers de logs Apache bien formé, et de stocker toutes les informations dans une structure de données. | 1.1 |
| Lorsqu'une URL de destination commence par l'URL de base du serveur, cette dernière doit être enlevée pour le calcul des statistiques. | 1.2 |
|  |  |
| **- Analyse des données -** | **2.\*** |
| Un fichier d'entrée contenant moins de 10 pages différentes est fourni, le programme doit être capable de les traiter. | 2.1 |
| Le programme doit être capable de n'afficher que les 10 pages les plus visitées (sauf en cas d'égalité entre certaines pages). | 2.2 |
| Lorsque l'option -e est utilisée , les fichiers de type image, script ou de formatage doivent être exclus du traitement. | 2.3 |
| L'option -g doit permettre de générer un fichier .dot bien formé, et correspondant aux données du fichier de logs. | 2.4 |

# Architecture

## Classes

Schéma général de l'application

|  |  |
| --- | --- |
| Log : | Cette classe permet de stocker toutes les informations extraites d'un log. Aucune information n'est traitée dans cette classe, tout est laissé brut, pour pouvoir être réutilisé par la suite, par n'importe quelle classe de calcul. Mis à part le constructeur par défaut, le constructeur de copie et le destructeur, les seules méthodes définies dans Log sont les opérateurs << et >> , pour faciliter la lecture à partir d'un fichier, et pour vérifier que les informations sont lues correctement.  Ses attributs sont toutes les informations contenues dans un log Apache.  La classe Log est déclarée friend dans Analyser, pour faciliter le traitement des logs. |
| Analyser : | Il s'agit de la classe qui gère le traitement des informations tirées des logs, au travers des méthodes DisplayTenMostVisited (qui affiche les 10 pages les plus visitées)et GenerateGraphViz (qui génère un fichier .dot permettant de créer une image grâce au programme GraphViz).  Son seul attribut est une liste de Log, qui est remplie à la construction de l'analyseur, à partir d'un fichier.  L'opérateur << a également été surchargé pour afficher un par un tous les logs de sa liste. |
| Launcher : | Launcher est la classe principale du projet. C'est elle qui appelle toutes les autres classes ainsi que leurs méthodes. Elle se charge également de vérifier que les paramètres donnés à l'appel du programme sont corrects.  Elle contient également le manuel du programme. |

## Structures de données

Les logs sont stockés sous la forme d'une liste dans Analyser, car ils doivent tous être lus par les méthodes de calcul. Il n'y a pas de souci d'accès rapide à un log particulier. De plus, une liste est performante en insertion, ce qui se révèle utile lors de la lecture du fichier de logs.

Par ailleurs, pour les besoins de la méthode DisplayTenMostVisited, nous avons défini une structure StringIntPair, qui permet d'associer une URL avec son nombre de visites. Nous l'utilisons dans une liste qui sera ensuite triée, grâce à la redéfinition des opérateurs < et ==, que nous n'aurions pas pu surcharger avec une pair de la STL.

Enfin, pour la méthode GenerateGraphViz, nous avions besoin d'associer à chaque couple (URL d'origine, URL de destination), le nombre de voyages. A cet effet, nous avions besoin d'une clé composée de ces deux URLs pour y associer un entier via la classe map de la STL. Nous avons donc créé une structure StringPair, dans laquelle sont redéfinis les opérateurs < et ==. La classe map de la STL permet d'accéder à des éléments avec une performance logarithmique en fonction du nombre d'éléments, ce qui est intérressant car nous avions besoin de faire de nombreux accès/insertions dans cette structure.

# Manuel

Ce manuel est également disponible sous la même forme en ligne de commande en appelant le programme sans aucun paramètre.

---- TP\_STL manual ----

Apache log file analyser

This program allows you to display the top ten most visited pages

of your apache server. The computation is done by providing your

apache log file.

For consistent results you should have a TP\_STL.cfg file in your

current directory containing in one line the base url of your server.

For instance, if your server is: http://intranet-if.insa-lyon.fr/%%%%

The file should contain http://intranet-if.insa-lyon.fr

Several other options are listed below.

USAGE: ./TP\_STL [OPTIONS]... FILE

OPTIONS:

-e :exclude the javascript, css an image files from the analysis,

focusing the computations on the other files.

-t <hour> :focus the computations on the logs that took place in the

[hour, hour+1] interval of time.

The hour should be an integer in the range [0,23]

-g <file> :generate a graphviz file describing the travels from

a page to another. file is the name of the file in

which the graph should be stored

If it exists already, the file will be overwritten.

FILE:

The file to analyse should be the last one to be given as parameter.

It has to be an APACHE log format file.