

Compte Rendu TPCpp3

Benoit RENAULT, Clément ESPEUTE

Décembre 2015

1 Introduction

Le programme **analog** permet de lire et d'extraire certaines informations intéressantes d'un fichier de log apache. Il permet entre autre de faire la liste des documents les plus consultés sur le serveur et d'établir un graphe des liens entre les différentes pages du site et de l'Internet et combien de fois ceux-ci ont été suivis. Le programme gère différents paramètres permettant par exemple d'ignorer certains types de fichier ou de ne prendre en compte que les requêtes qui ont eu lieu à une heure donnée.

2 Spécifications

Spécification	Test(s)
Ouverture des fichiers Le programme tentera toujours d'ouvrir le fichier correspondant à son dernier argument fourni en paramètre comme le fichier le log à lire. Si ce fichier n'existe pas ou qu'il ne peut pas être ouvert, alors un message d'erreur apparaîtra dans la sortie erreur standard.	Tout les tests utilisent le fait de pouvoir ouvrir un fichier pour fonctionner et donc permettent de vérifier ce celle-ci fonctionne. TestFichierExistePas s'occupe lui de vérifier que le programme affiche bien le message d'erreur en cas d'ouverture raté d'un fichier qui n'existe, et TestFichierEcritureSeule teste le cas ou le fichier n'est pas disponible en lecture.
Lecture d'une ligne de log : Chaque ligne du log Apache doit être lue une par une pour en extraire les informations intéressantes. Toutes les lignes doivent suivre le même format : 192.168.0.0 - - [JJ/mois/AAAA :HH :MM :SS +GMT] "GET adressePageDemandée protocole" codeRetourHttp nombre "adresseDeProvenanceDeLaRequête" "userAgent". Les lignes qui ne suivent pas ce format seront simplement ignorées sans message d'erreur. En particulier, seule les requêtes utilisant la méthode GET seront lues par le log. De plus, certaines parties de adressePageDemandée ne seront pas prise en compte. Cela inclus les données complémentaires des pages php ou js (celles situées après le ? ou le ;)	TestLireLog s'occupe de vérifier que chaque ligne du log est bien lue. On vérifie notamment que chaque partie de chaque ligne de log est bien découpée selon les données qui doivent être comprises. Comme il est impossible de prévoir tous les cas possibles dans ces tests, on mets en place TestNombreLectures qui utilise un très gros fichier de log fourni lors du TP. Le but est de comparer le nombre de lignes lues par le programme et le nombre de lignes contenues dans le fichier (note, seulement celles qui contiennent un GET sont prises en compte). Si le test passe, cela signifie que l'on a bien réussi à capturer l'ensemble des ligne du fichier log. On ne teste pas en revanche si les données sont bien lues, car il faudrait lire ce fichier à la main ou à l'aide d'un autre programme, ce qui impliquerait des possibilités d'erreurs dans tous les cotes.

<p>Affichage des 10 documents les plus consultés Par défaut, le programme ne fait qu'afficher cette liste. Elle prend le format <i>nomDocument (nbre_vue hits)</i></p>	<p>TestClassementVisite s'occupe de vérifier que la liste des 10 premiers documents visités correspond bien aux visites du log, et que le format est bien respecté.</p>
<p>Création du fichier GraphViz En passant l'argument <i>-g nomfichier.dot</i> on peut générer un fichier GraphViz correspondant aux liens suivis entre chaque document du site. Par défaut les sites externes sont tous regroupés dans un document "Externe". On peut choisir de les afficher en passant le paramètre <i>-x</i> au programme. Dans ce cas, il apparaîtront sous la forme <i>externe : nom_du_site</i></p>	<p>Comme il est difficile de prévoir dans quel ordre les documents apparaîtront dans le fichier GraphViz, il nous est impossible de réaliser un test automatisé pour cette spécification. Ceux-ci seront uniquement fait de manière visuelle.</p>
<p>Exclusion de certaines extensions. En passant l'argument <i>-e</i> au programme, on peut choisir d'ignorer tous les documents ayant pour extension <i>png,gif,jpg,jpeg,ico,css,js</i>. Toute requête pointant vers un de ces documents sera simplement ignorée. En revanche, les requêtes provenant de ces sites seront prise en compte.</p>	<p>TestIgnorerExtensions s'occupe de vérifier que l'ensemble des extensions prévues dans la spécification sont ignorées.</p>
<p>Créneau Horaire. En passant <i>-i heure</i> on peut décider de ne récupérer que les logs qui ont eu lieu à une heure donnée. Toutes les autres requêtes sont ignorées.</p>	<p>TestIgnorerHeure s'occupe de vérifier que l'on ignore bien les logs qui n'arrivent pas à une heure donnée.</p>
<p>Changer le site étudié Par défaut, le programme considère toutes les requêtes provenant de <i>intranet-if.insa-lyon.fr</i> comme étant des requêtes internes, et toutes les autres comme étant des requêtes externes. Si l'on veut changer ce comportement, on peut utiliser l'argument <i>-x nomDuSite</i>. Il est à la charge de l'utilisateur de fournir une adresse de base correcte de la forme <i>sousDomaine.domaine.extension</i>, sans quoi toutes les requêtes seront considérées comme externes.</p>	<p>TestChangerSite s'occupe de vérifier que l'on obtiens toujours des résultats même lorsque l'on change l'adresse du site d'origine</p>
<p>Autres arguments <i>-v</i> permet d'afficher l'ensemble des données lues et comprises par le programme. Cela est utile pour l'élaboration des tests. <i>-q</i> permet de ne pas afficher la liste des 10 premiers documents visités, encore une fois pour faciliter les tests. <i>-h</i> permet d'afficher l'aide du programme</p>	<p>Aucun, mais ces arguments sont utiles pour les autres tests.</p>

Note sur les tests : La plupart des tests portant sur les fichiers sont générés aléatoirement à l'aide d'un script lua. Ce script est chargé de créer un fichier de log et un fichier de sortie qui couvrent la plus grande étendue des cas possibles. A chaque fois, le script se nomme **générerFichier.lua** et peut être invoqué en appelant le script **genererNouveauTest.sh** dans chaque dossier de Tests, ou simplement en appelant

`rebuildTests.sh` dans le dossier `Tests`.

3 Choix des structures de données

4 Conception (Diagramme de classe)

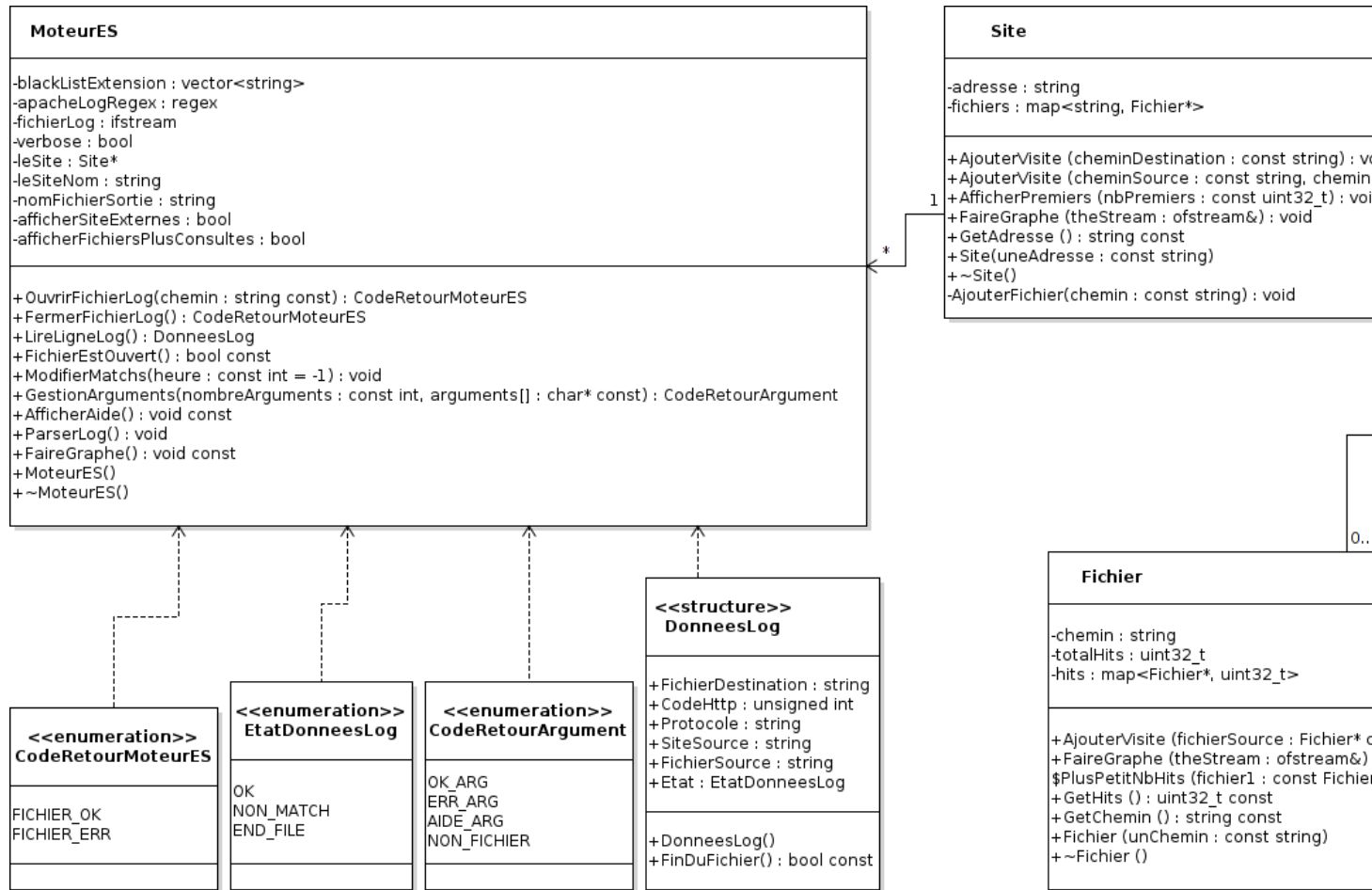


FIG. 1 : Caption