

Manuel d'Utilisation : Analog

Analyseur de Logs Apache

Félix PIETRUCZANIS, Raphaël GUTEVILLE,
Jules BRISARD, Gerald CHAMBI RAMOS

Février 2026

Table des matières

1	Introduction	2
2	Installation et Compilation	2
2.1	Compilation du projet	2
2.2	Nettoyage	2
3	Guide de Configuration (config.txt)	2
4	Utilisation	2
4.1	Options disponibles	3
4.2	Comportement par défaut	3
5	Visualisation du Graphe	3
6	Spécifications Fonctionnelles et Tests	3
6.1	Tableau des Spécifications	3
6.2	Plan de Tests et Framework	3
7	Détails Techniques	3
8	Exemples de Sorties et Résultats Attendus	4
8.1	Analyse Standard (Top 10)	4
8.2	Analyse avec Filtrage et Génération de Graphe	4
8.3	Filtrage Horaire	5

1 Introduction

Analog est un outil développé en C++ conçu pour l'analyse syntaxique et statistique des fichiers journaux (logs) de serveurs web Apache. Il permet non seulement d'extraire les documents les plus consultés, mais aussi de visualiser le parcours des utilisateurs sous forme de graphes orientés.

2 Installation et Compilation

Le projet est fourni avec un **Makefile** pour automatiser la gestion des dépendances et appliquer les optimisations nécessaires.

2.1 Compilation du projet

Ouvrez un terminal dans le répertoire racine du projet et exécutez :

```
1 make
```

2.2 Nettoyage

Pour supprimer les fichiers objets (.o) et l'exécutable après utilisation, tapez :

```
1 make clean
```

3 Guide de Configuration (config.txt)

Le comportement de l'analyseur est piloté par le fichier **config.txt**. Ce fichier définit les paramètres locaux de votre environnement vous pouvez les modifier selon votre utilisation.

- **base_url** : L'URL racine considérée comme locale (par défaut : `http://intranet-if.insa-lyon.fr`).
- **file_ext** : La liste des extensions à exclure lors de l'utilisation de l'option **-e** (par défaut : les images, documents CSS et JS).

Le fichier **config.txt** est essentiel pour adapter l'outil à votre environnement serveur.

- **Modifier l'URL locale** : Changez la valeur de **base_url** pour que le programme puisse identifier et nettoyer les referers internes.
- **Ajouter des extensions à exclure** : Pour exclure de nouveaux types de fichiers (comme le format **.webp**), ajoutez simplement **"webp"** à la liste **file_ext** entre guillemets et séparé par une virgule.

Attention

Le programme recherche le fichier **config.txt** dans le répertoire d'exécution courant. Assurez-vous qu'il est présent avant de lancer l'analyse.

4 Utilisation

La syntaxe générale de la commande à entrer dans le terminal est la suivante :

```
1 ./analog [-g nomfichier.dot] [-e] [-t heure] nomfichier.log
```

Option	Description
-g <file.dot>	Génère un fichier au format GraphViz (.dot) représentant les flux entre les pages.
-e	Exclut les ressources définies dans la configuration.
-t <heure>	Filtre les requêtes pour ne garder que celles effectuées dans le créneau [<i>heure</i> , <i>heure</i> + 1[.

TABLE 1 – Options de la ligne de commande

4.1 Options disponibles

4.2 Comportement par défaut

Sans option particulière, **Analog** analyse l'intégralité du fichier de log et affiche le **Top 10** des documents les plus consultés par ordre décroissant de popularité.

5 Visualisation du Graphe

Si vous avez utilisé l'option **-g**, vous obtiendrez un fichier **.dot**. Pour convertir ce fichier en image, vous devez disposer de l'outil **Graphviz**.

Exécutez la commande suivante pour générer un fichier PNG :

```
1 dot -Tpng mon_graphe.dot -o graphe.png
```

Conseil

Si votre graphe contient de nombreux nœuds, privilégiez le format SVG (**-Tsvg**) pour pouvoir zoomer sans perte de qualité.

6 Spécifications Fonctionnelles et Tests

Conformément aux exigences du projet, cette section détaille les spécifications du programme **analog** et la stratégie de test employée pour valider chaque fonctionnalité.

6.1 Tableau des Spécifications

Le tableau suivant récapitule les comportements attendus du logiciel pour les cas nominaux et les cas d'erreur identifiés dans la conception.

6.2 Plan de Tests et Framework

Pour valider ces spécifications, nous utilisons le framework de test fourni (scripts shell) qui compare les sorties produites avec des sorties de référence. Chaque test est isolé dans un répertoire spécifique.

7 Détails Techniques

L'application repose sur plusieurs classes clés :

- **LogReader** : Responsable de l'ouverture et du parsing ligne à ligne du fichier log.
- **LogStats** : Gère le stockage en mémoire (via **unordered_map**) et le calcul des fréquences.
- **GraphMaker** : Transforme les données d'interactions en syntaxe DOT.

Condition / Entrée	Comportement Attendu
Exécution sans option : <code>./analog court.log</code>	Affiche le Top 10 des documents les plus consultés par ordre décroissant.
Option <code>-g <file.dot></code> active	Génère un fichier au format GraphViz représentant les arcs (referer → cible) et leurs poids.
Option <code>-e</code> active	Exclut de l'analyse les types de fichiers (images, css, js) définis dans <code>config.txt</code> .
Option <code>-t <heure></code> active	Ne comptabilise que les requêtes dont l'heure est dans $[heure, heure + 1[$.
Fichier de log inexistant ou inaccessible	Affiche un message d'erreur sur <code>stderr</code> et interrompt l'exécution.
Heure invalide pour <code>-t</code> (hors 0-23)	Affiche une erreur indiquant que l'heure doit être un entier entre 0 et 23.
Nom de fichier <code>.dot</code> manquant ou invalide	Affiche un message d'erreur et demande un nom de fichier correct pour l'option <code>-g</code> .

TABLE 2 – Spécifications fonctionnelles et de gestion d'erreurs

8 Exemples de Sorties et Résultats Attendus

Cette section présente ce que l'utilisateur doit observer dans son terminal lors de l'exécution des commandes principales.

8.1 Analyse Standard (Top 10)

Lorsqu'aucune option n'est spécifiée, le programme affiche uniquement les dix documents les plus consultés par ordre décroissant de hits.

Listing 1 – Sortie console par défaut

```

1 $ ./analog test.log
2 Classement des 10 documents les plus consultes :
3 /page2.html (3 hits)
4 /image.jpg (1 hits)
5 /page3.html (1 hits)
6 /page1.html (1 hits)
```

8.2 Analyse avec Filtrage et Génération de Graphe

L'utilisation combinée des options `-e` et `-g` permet d'exclure les ressources statiques et de produire le fichier pour GraphViz.

Listing 2 – Sortie avec filtrage et graphe

```

1 $ ./analog -e -g mon_graphe.dot test.log
2 Les fichiers images, CSS et JS seront exclus de l'analyse.
3 Classement des 10 documents les plus consultes :
4 /page2.html (3 hits)
5 /page3.html (1 hits)
6 /page1.html (1 hits)
7 Fichier .dot genere avec succes.
```

Note : L'exclusion se base sur les extensions définies dans `config.txt` (par défaut : png, jpg, css, js).

8.3 Filtrage Horaire

L'option `-t` restreint l'analyse à un créneau d'une heure précise.

Listing 3 – Sortie avec filtrage horaire

```
1 $ ./analog -t 12 test.log
2 Attention : seulement les hits entre 12h et 13h sont pris en compte.
3 Classement des 10 documents les plus consultes :
4 /page2.html (2 hits)
5 /image.jpg (1 hits)
6 /page3.html (1 hits)
```

Note : Le créneau est calculé selon l'intervalle $[heure, heure + 1[$.