

TP PHP : Analyse de Logs Serveur

Mission

Vous êtes administrateur système chez **CloudMaster**. Un client se plaint de lenteurs intermittentes. Il vous a envoyé un extrait brut des logs de son serveur Apache.

Ces logs sont séparés en deux tableaux PHP distincts (mais synchronisés par leur index) :

1. `$httpCodes` : Les codes de statut HTTP (200 = OK, 404 = Introuvable, 500 = Erreur Serveur).
2. `$responseTimes` : Le temps mis par le serveur pour répondre (en millisecondes).

Votre objectif : Créer un script PHP en ligne de commande (CLI) pour auditer ces données et sortir des métriques précises.

Les Données

Créez un fichier **monitoring.php** et collez ces données tout en haut :

```
<?php
// =====
// DONNÉES BRUTES (LOGS DU 01/12/2023)
// =====

// Liste des codes HTTP renvoyés par le serveur
$httpCodes = [200, 200, 500, 200, 200,
              404, 200, 500, 500, 200,
              200, 200, 404, 200, 200
];

// Liste des temps de réponse correspondants (en ms)
// L'index 0 de $responseTimes correspond à l'index 0 de $httpCodes, etc.
$responseTimes = [120, 45, 1500, 600, 120,
                  40, 2500, 300, 800, 95,
                  110, 450, 20, 3500, 150
];
```

Partie 1 : Vue d'ensemble (Synchronisation)

Avant d'analyser, il faut visualiser.

1. Affichez le nombre total de requêtes analysées.
2. Affichez l'historique complet ligne par ligne sous ce format :
Requête [0] : Code 200 | Temps : 120 ms
Requête [1] : Code 200 | Temps : 45 ms
...

Attention : Les deux tableaux ont la même taille. L'index vous servira à piocher dans les deux tableaux en même temps.

Partie 2 : Détection d'incidents (Recherche d'extremums)

Votre client veut savoir quelle a été la requête la plus lente de la série pour l'investiguer.

1. Trouvez le temps de réponse **le plus élevé** (max()).
2. Trouvez l'index (la position) de cette requête dans le tableau.
3. Affichez les détails de cet incident :
 - "La requête la plus lente a mis X ms."
 - "Elle correspond à l'index N."
 - "Son code HTTP était : Y."

Partie 3 : Audit de performance

1. Taux d'erreur :

Parcourez le tableau `$httpCodes` pour compter combien de fois l'erreur 500 est apparue. Affichez ce nombre.

2. Analyse conditionnelle (Le défi) :

Calculer la moyenne générale est facile (`array_sum / count`). Mais cela ne nous intéresse pas car les erreurs (souvent rapides ou très lentes) faussent tout.

Nous voulons la moyenne des temps de réponse **UNIQUEMENT** pour les requêtes valides (Code 200).

Algorithme à suivre :

- Initialisez une variable pour la somme des temps (`$sommeValide`).
- Initialisez une variable pour compter le nombre de requêtes valides (`$compteurValide`).
- Parcourez les tableaux.
- Si le code est 200 -> Ajoutez le temps à la somme ET incrémentez le compteur.
- À la fin, calculez la moyenne (`Somme / Compteur`).

Partie 4 : Le rapport d'alerte

Une requête est considérée comme **critique** si elle remplit DEUX conditions :

1. Elle est en succès (Code 200).
2. Elle est anormalement lente (> 500 ms).

Consigne :

Parcourez les données et affichez uniquement les requêtes qui posent problème.

Format de sortie : ALERTE : La requête [Index] est valide mais trop lente (Temps ms) !

Bonus

Affichez la répartition des codes HTTP en pourcentage.

Exemple :

- Code 200 : X %
- Code 404 : Y %
- Code 500 : Z %

Exemple d'exécution

RAPPORT DE MONITORING (CORRECTION)

[INFO] Nombre de logs analysés : 15

--- 1. HISTORIQUE DES REQUÊTES ---

Requête [0] : Code 200 | Temps : 120 ms
Requête [1] : Code 200 | Temps : 45 ms
Requête [2] : Code 500 | Temps : 1500 ms
Requête [3] : Code 200 | Temps : 600 ms
Requête [4] : Code 200 | Temps : 120 ms
Requête [5] : Code 404 | Temps : 40 ms
Requête [6] : Code 200 | Temps : 2500 ms
Requête [7] : Code 500 | Temps : 300 ms
Requête [8] : Code 500 | Temps : 800 ms
Requête [9] : Code 200 | Temps : 95 ms
Requête [10] : Code 200 | Temps : 110 ms
Requête [11] : Code 200 | Temps : 450 ms
Requête [12] : Code 404 | Temps : 20 ms
Requête [13] : Code 200 | Temps : 3500 ms
Requête [14] : Code 200 | Temps : 150 ms

--- 2. INCIDENT LE PLUS CRITIQUE ---

Temps record : 3500 ms
Index concerné : 13
Code HTTP : 200 (C'est le code associé à ce temps)

--- 3. STATISTIQUES AVANCÉES ---

Nombre d'erreurs 500 : 3
Temps moyen (Succès uniquement) : 769 ms

--- 4. ALERTES (Faux Positifs) ---

⚠ ALERTE : La requête [3] est valide mais LENTE (600 ms)
⚠ ALERTE : La requête [6] est valide mais LENTE (2500 ms)
⚠ ALERTE : La requête [13] est valide mais LENTE (3500 ms)

--- BONUS : RÉPARTITION ---

Code HTTP 200 : 66.7 % des requêtes.
Code HTTP 500 : 20 % des requêtes.
Code HTTP 404 : 13.3 % des requêtes.

Fin du rapport.