

Nom : Loubna MANGOUCI
N°Étudiant : 124099931
Classe : INFO3

TP4 – CouchDB et le Paradigme MapReduce

1. Présentation de CouchDB

CouchDB, développé par la fondation Apache, est une base de données NoSQL open-source orientée documents. Elle a été conçue pour faciliter la réplication, la synchronisation et la disponibilité des données, même dans des environnements distribués. Les informations sont stockées nativement au format JSON, ce qui la rend particulièrement adaptée aux applications modernes.

2. Le Modèle Orienté Documents

Contrairement aux bases relationnelles classiques, CouchDB repose sur un modèle documentaire flexible. Les données sont organisées sous forme de documents JSON indépendants, chacun étant identifié par un champ unique `_id`.

Caractéristiques principales :

- **JSON natif** : Les données sont stockées sous forme clé-valeur, faciles à lire et à manipuler.
- **Souplesse du schéma** : Les documents d'une même base peuvent avoir des structures différentes.
- **Autonomie des documents** : Il n'existe pas de relations rigides entre les documents, ce qui simplifie la réplication.

Exemple de document :

```
{  
  
  "_id": "001",  
  
  "name": "Alice",  
  
  "age": 25,  
  
  "city": "Paris"  
  
}
```

3. Architecture et Fonctionnement de CouchDB

Le fonctionnement de CouchDB repose sur plusieurs concepts fondamentaux :

- **Interface REST** : Toutes les opérations s'effectuent via des requêtes HTTP standards (GET, POST, PUT, DELETE).
- **Réplication maître-maître** : Les bases peuvent se synchroniser dans les deux sens sans serveur central.
- **Vues MapReduce** : Utilisées pour indexer les données et effectuer des requêtes avancées.
- **Garanties ACID** : Assurées au niveau de chaque document.
- **Gestion automatique des conflits** : Les conflits sont détectés et traités lors de la réplication.

4. MapReduce avec CouchDB

4.1 Principe du Paradigme MapReduce

MapReduce est un modèle de traitement distribué divisé en deux phases :

1. **Map** : Analyse chaque document et génère des paires clé–valeur.
2. **Reduce** : Regroupe et agrège les valeurs associées à une même clé (somme, moyenne, etc.).

Ce modèle permet de traiter efficacement de grands volumes de données en parallèle.

4.2 Mise en œuvre dans CouchDB

Dans CouchDB, les fonctions Map et Reduce sont définies dans des Design Documents, ce qui permet de créer des vues persistantes et indexées.

Exemple : calcul d'un stock total

- **Fonction Map** : Émet le nom du produit et la quantité correspondante.

```
function(doc) { emit(doc.produit, doc.quantite); }
```

- **Fonction Reduce** : Calcule la somme des quantités.

```
function(keys, values) { return sum(values); }
```

5. Utilisation de l'API REST

CouchDB peut être utilisé sans outil spécifique grâce à son API REST. Les opérations courantes peuvent être effectuées à l'aide de la commande **curl**.

Action	Exemple de commande

Vérifier le serveur	<code>curl -X GET http://localhost:5984/</code>
Créer une base	<code>curl -X PUT http://localhost:5984/nom_db</code>
Ajouter un document	<code>curl -X PUT http://localhost:5984/nom_db/id_doc -d '{"key": "val"}'</code>
Insertion multiple	Utilisation de l'endpoint <code>_bulk_docs</code>

6. Déploiement de CouchDB avec Docker

Docker permet de lancer rapidement une instance CouchDB sans configuration complexe :

```
docker run -d --name couchdb -p 5984:5984 \
-e COUCHDB_USER=admin -e COUCHDB_PASSWORD=password \
couchdb
```

Une fois le conteneur démarré, l'interface graphique Fauxton est accessible à l'adresse : http://localhost:5984/_utils/

Exercice 1 – Calculs Matriciels avec MapReduce

1. Représentation de la Matrice M

Dans le cadre d'un calcul de type PageRank, une matrice $N \times N$ est modélisée en stockant chaque ligne sous forme d'un document JSON.

```
{  
  
  "page": "P_i",  
  
  "links": [  
  
    { "target": "P_j1", "weight": 0.5 },  
  
    { "target": "P_j2", "weight": 0.3 }  
  
  ]  
  
}
```

2. Calcul de la Norme d'un Vecteur

La norme d'un vecteur V est définie par :

$$|V| = \sqrt{(v_1^2 + v_2^2 + \dots + v_n^2)}$$

- **Map** : Pour chaque lien, on émet le carré du poids.
- **Reduce** : On additionne tous les carrés, puis on applique la racine carrée au résultat final.

3. Produit de la Matrice M par le Vecteur W

Le produit est donné par la formule :

$$\varphi_i = \sum_{j=1}^N M_{ij} \cdot w_j$$

- **Hypothèse** : Le vecteur W est disponible en mémoire.

- **Map** : Chaque poids M_i est multiplié par la valeur correspondante w_i .
- **Reduce** : Les produits sont additionnés afin d'obtenir la valeur finale ϕ_i pour chaque page.

Conclusion sur les Performances

L'approche MapReduce permet de répartir efficacement les calculs sur plusieurs nœuds. Cette distribution rend possible le traitement de matrices de grande taille en tirant pleinement parti du parallélisme, ce qui améliore significativement les performances globales.