



Université de Nouakchott
Faculté des Sciences et Techniques
Département Informatique



Rapport MongoDB et Hadoop

Réalisé par :
Houssein Hmoudi M1-SI-C20536

17 juin 2025

Table des matières

1	Introduction générale	2
2	Partie I : TD N°1 - Introduction à MongoDB	2
2.1	Environnement de Travail	2
2.2	Exercice 1 : Gestion de la collection produits	2
2.2.1	Sélection de la base	2
2.2.2	Insertion de documents	2
2.2.3	Requêtes de lecture	3
2.2.4	Suppression de documents	3
2.3	Conclusion MongoDB	3
3	Partie II : Projet Hadoop - Compteur de mots (WordCount)	3
3.1	Contexte du Projet	3
3.2	Environnement Technique	4
3.3	Étapes Clés	4
3.3.1	a) Installation et Configuration de Hadoop	4
3.3.2	b) Configuration de SSH	4
3.3.3	c) Initialisation de HDFS et YARN	4
3.3.4	d) Développement du Job MapReduce	4
3.3.5	e) Exécution du Job	4
3.4	Résultats et Validation	5
3.5	Conclusion Hadoop	5
4	Conclusion Générale	5

1 Introduction générale

Ce rapport combine deux travaux pratiques réalisés dans le cadre des enseignements de Systèmes de Gestion de Bases de Données NoSQL et de traitements distribués : le TD N°1 sur MongoDB et le projet Hadoop WordCount. L'objectif est de présenter de manière cohérente les objectifs, méthodes et résultats des deux expériences, soulignant les compétences acquises en gestion de données et en programmation distribuée.

2 Partie I : TD N°1 - Introduction à MongoDB

2.1 Environnement de Travail

- **Système d'exploitation** : Windows
- **Version MongoDB** : 7
- **Outil** : mongo (MongoDB Shell)
- **Base de données ciblée** : info

2.2 Exercice 1 : Gestion de la collection produits

2.2.1 Sélection de la base

```
1 use info
2 // Vérifier : db "info"
```

2.2.2 Insertion de documents

```
1 // MacBook Pro
2 db.produits.insertOne({
3     nom: "Macbook Pro",
4     fabricant: "Apple",
5     prix: 17435.99,
6     options: ["Intel Core i5", "Retina Display", "Long life battery"]
7 })
8
9 // Autres produits
10 db.produits.insertMany([
11     {
12         nom: "DELL",
13         fabricant: "Dell Technologies, Inc",
14         prix: 1143,
15         options: ["Intel Core Ultra 9 285H", "SSD", "32 Go"]
16     },
17     {
18         nom: "Thinkpad X230",
19         fabricant: "Lenovo",
20         prix: 114358.74,
21         ultrabook: true,
22         options: ["Intel Core i5", "SSD", "Long life battery"]
23     }
24 ])
```

2.2.3 Requêtes de lecture

```
1 // Tous les produits
2 db.produits.find().pretty()
3
4 // Premier produit
5 db.produits.findOne()
6
7 // Par _id
8 let id = db.produits.findOne({ nom: "Thinkpad X230" })._id;
9 db.produits.findOne({ _id: id });
10
11 // Prix > 13 723 DA
12 db.produits.find({ prix: { $gt: 13723 } }).pretty()
13
14 // Ultrabook
15 db.produits.findOne({ ultrabook: true })
16
17 // Nom contenant / commen ant par "Macbook"
18 db.produits.find({ nom: /^Macbook/ }).pretty()
```

2.2.4 Suppression de documents

```
1 // Par fabricant
2 db.produits.deleteOne({ fabricant: "Lenovo" })
3
4 // Par id
5 db.produits.deleteOne({ _id: id })
```

2.3 Conclusion MongoDB

Ce TP a permis de maîtriser :

- La création et sélection de bases/collections
- Les opérations CRUD de base
- L'utilisation de filtres et d'expressions régulières

Les prochaines étapes incluront la mise à jour de documents, l'optimisation par index et l'usage du pipeline d'agrégation.

3 Partie II : Projet Hadoop - Compteur de mots (Word-Count)

3.1 Contexte du Projet

L'objectif de ce projet est d'implémenter l'algorithme MapReduce WordCount sur un cluster Hadoop mono-nœud afin de traiter un fichier texte (`poeme.txt`) et de générer la fréquence d'apparition de chaque mot.

3.2 Environnement Technique

Composant	Version / Détails
Système	Ubuntu 24.04 (VirtualBox)
Hadoop	3.2.1
Java	OpenJDK 11

3.3 Étapes Clés

3.3.1 a) Installation et Configuration de Hadoop

```
1 wget https://archive.apache.org/dist/hadoop/common/hadoop-3.2.1/hadoop-3.2.1.tar.gz
2 tar -xzf hadoop-3.2.1.tar.gz
3 sudo mv hadoop-3.2.1 /usr/local/hadoop
4
5 # Variables d'environnement dans ~/.bashrc
6 export HADOOP_HOME=/usr/local/hadoop
7 export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
8 export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
```

3.3.2 b) Configuration de SSH

```
1 ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa
2 cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
3 chmod 600 ~/.ssh/authorized_keys
4 chmod 700 ~/.ssh
```

3.3.3 c) Initialisation de HDFS et YARN

```
1 hdfs namenode -format
2 start-dfs.sh
3 start-yarn.sh
```

3.3.4 d) Développement du Job MapReduce

- **Mapper** : tokenisation du texte et émission de paires (mot, 1)
- **Reducer** : agrégation des occurrences

```
1 # Compilation
2 mkdir classes
3 javac -classpath $(hadoop classpath) -d classes WCount.java
4 jar -cvf tp-1.0.0.jar -C classes/ .
```

3.3.5 e) Exécution du Job

```
1 hadoop fs -put poeme.txt /
2 hadoop jar tp-1.0.0.jar WCount /poeme.txt /results2
```

3.4 Résultats et Validation

```
1 hadoop fs -ls /results2
2 hadoop fs -cat /results2/part-r-00000
3 # hadoop 3
4 # big 1
5 # data 2
6 # ...
```

3.5 Conclusion Hadoop

Ce projet a démontré :

- La configuration d'un cluster Hadoop mono-nœud
- L'implémentation réussie d'un algorithme MapReduce
- Le traitement distribué d'un corpus textuel

4 Conclusion Générale

La réalisation de ces deux travaux pratiques a renforcé les compétences en manipulation de bases de données NoSQL et en programmation distribuée. D'une part, MongoDB a offert une approche flexible pour le stockage et l'interrogation de données semi-structurées. D'autre part, Hadoop a permis de comprendre les principes de traitement parallèle sur de grands volumes de données.