

Rapport détaillé l'évolution du projet Hadoop

Introduction

Le projet avait pour but d'utiliser Hadoop et MapReduce pour traiter les données de films, en analysant des fichiers tels que `movies.csv` et `ratings.csv`. L'objectif était de répondre à plusieurs questions avancées, impliquant des jointures et des comptages complexes, et d'en apprendre davantage sur le traitement distribué des données à grande échelle.

Les étapes ont été réalisées en travaillant dans le répertoire suivant :

```
hadoop-first-code\src\main\java\org\hadoop\examples\hadoop\project
```

Structure et environnement

Arborescence du projet

```
project
├── driver
├── mapper
└── reducer
```

Configuration de l'environnement

1. Cluster Hadoop avec Docker :

- Un fichier `docker-compose.yml` a été utilisé pour lancer un cluster Hadoop comprenant les services suivants : `namenode`, `datanode`, `resourcemanager`, `nodemanager`.
- Le réseau Docker a été configuré pour éviter les conflits (`hadoop`).

2. Commandes pour configurer le cluster :

- Nettoyer les services existants :

```
docker-compose down -v
```

- **Lancer le cluster :**

```
docker-compose up -d
```

- **Vérifier les services :**

```
docker ps
```

3. Accès aux conteneurs :

```
docker exec -it namenode /bin/bash
```

1ère étape : Trouver le film le mieux noté par utilisateur

Objectif

Identifier le **movieId** ayant obtenu la note maximale pour chaque utilisateur.

Mise en œuvre

Mapper : SKHighestRatedMovieMapper

- Lit chaque ligne de **ratings.csv** .
- Émet **(userId, movieId:rating)** .

Reducer : SKHighestRatedMovieReducer

- Trouve le film avec la note maximale pour chaque utilisateur.

Driver : SKHighestRatedMovieDriver

- Configure le job Hadoop pour mapper et réduire les données.

Commandes

1. Charger les fichiers dans HDFS :

```
hdfs dfs -put /path/to/ratings.csv /input_ml25m/ratings.csv
```

2. Exécuter le job :

```
hadoop jar /tmp/hadoop-first-code-1.0-SNAPSHOT.jar org.hadoop.examples.hadoop.project.driver.SKHighestRatedMovieDriver /input_ml25m/ratings.csv /output_highest_rate  
d_movie_user
```

3. Résultats :

```
hdfs dfs -cat /output_highest Rated_movie_user/part-*
```

Résultat (extrait)

```
99985    1 (Rating: 5.0)  
99986   2395 (Rating: 5.0)  
99987    1 (Rating: 5.0)
```

```
99961   318 (Rating: 5.0)  
99962    16 (Rating: 5.0)  
99963   2959 (Rating: 5.0)  
99964    527 (Rating: 5.0)  
99965    16 (Rating: 5.0)  
99966    50 (Rating: 5.0)  
99967   1372 (Rating: 5.0)  
99968    11 (Rating: 5.0)  
99969    527 (Rating: 5.0)  
9997    26366 (Rating: 5.0)  
99970    260 (Rating: 5.0)  
99971   1246 (Rating: 4.5)
```

2ème étape : Jointure pour obtenir les titres des films

Objectif

Associer les titres des films à chaque utilisateur ayant donné une note maximale.

Mise en œuvre

Mapper 1 : SKMoviesMapper

- Lit `movies.csv` .
- Émet `(movieId, MOVIE:movieTitle)` .

Mapper 2 : SKUserRatingsMapper

- Lit les résultats de la 1ère étape.
- Émet `(movieId, USER:userId)` .

Reducer : SKMoviesJoinReducer

- Combine les données pour produire `(userId, movieTitle)` .

Driver : SKMoviesJoinDriver

- Configure le job Hadoop pour mapper et réduire les données.

Commandes

1. Charger les fichiers dans HDFS :

```
hdfs dfs -put /tmp/movies.csv /input_ml25m/movies.csv
hdfs dfs -put /output_highest_rated_movie_user /input/user_ratings.csv
```

```
root@0676b262fa85:/# ^C
root@0676b262fa85:/# hdfs dfs -get /output_highest_rated_movie_user/part-* /tmp/user_ratings.csv
2025-01-19 03:23:36,499 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localhostTrusted = false, remoteHostTrusted = false
root@0676b262fa85:/#
```

```
root@0676b262fa85:/# ^C
root@0676b262fa85:/# ^C
root@0676b262fa85:/# hdfs dfs -get /output_highest_rated_movie_user/part-* /tmp/user_ratings.csv
2025-01-19 03:23:36,499 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localhostTrusted = false, remoteHostTrusted = false
root@0676b262fa85:/# hdfs dfs -mkdir -p /input
root@0676b262fa85:/# hdfs dfs -put /tmp/user_ratings.csv /input/user_ratings.csv
2025-01-19 03:25:50,078 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localhostTrusted = false, remoteHostTrusted = false
root@0676b262fa85:/#
```

2. Exécuter le job :

```
hadoop jar /tmp/hadoop-first-code-1.0-SNAPSHOT.jar org.hadoop.examples.hadoop.project.driver.SKMoviesJoinDriver /input_ml25m/movies.csv /input/user_ratings.csv /output_movies_user_likes
```

```

root@0676b262fa85:/# hadoop jar /tmp/hadoop-first-code-1.0-SNAPSHOT.jar org.hadoop.examples.hadoop.project.driver.SKMoviesJoinDriver /input_ml25m/movies.csv /input/user_ratings.csv /output_movies_user_likes
2025-01-19 03:36:03,860 INFO client.RMProxy: Connecting to ResourceManager at resourcemanager/172.20.0.6:8032
2025-01-19 03:36:03,969 INFO client.AHSProxy: Connecting to Application History server at historyserver/172.20.0.4:10200
2025-01-19 03:36:04,077 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with Tool Runner to remedy this.
2025-01-19 03:36:04,095 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/root/.staging/job_1737254157188_0005
2025-01-19 03:36:04,162 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localhostTrusted = false, remoteHostTrusted = false
2025-01-19 03:36:04,242 INFO input.FileInputFormat: Total input files to process : 1
2025-01-19 03:36:04,256 INFO input.FileInputFormat: Total input files to process : 1
2025-01-19 03:36:04,278 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localhostTrusted = false, remoteHostTrusted = false
2025-01-19 03:36:04,734 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localhostTrusted = false, remoteHostTrusted = false
2025-01-19 03:36:04,749 INFO mapreduce.JobSubmitter: number of splits:2
2025-01-19 03:36:04,831 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localhostTrusted = false, remoteHostTrusted = false
2025-01-19 03:36:04,855 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1737254157188_0005
2025-01-19 03:36:04,855 INFO mapreduce.JobSubmitter: Executing with tokens: []
2025-01-19 03:36:04,991 INFO conf.Configuration: resource-types.xml not found
2025-01-19 03:36:04,991 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2025-01-19 03:36:05,266 INFO impl.YarnClientImpl: Submitted application application_1737254157188_0005
2025-01-19 03:36:05,293 INFO mapreduce.Job: The url to track the job: http://resourcemanager:8088/proxy/application\_1737254157188\_0005/
2025-01-19 03:36:05,293 INFO mapreduce.Job: Running job: job_1737254157188_0005
2025-01-19 03:36:11,385 INFO mapreduce.Job: Job job_1737254157188_0005 running in uber mode : false
2025-01-19 03:36:11,385 INFO mapreduce.Job: map 0% reduce 0%
2025-01-19 03:36:16,435 INFO mapreduce.Job: map 50% reduce 0%

```

3. Résultats :

```
hdfs dfs -cat /output_movies_user_likes/part-*
```

Résultat (extrait)

```

64695    Last Man Standing (1996)
8802     It's Such a Beautiful Day (2012)

```

```

158625   Big Night (1996)
33084    Big Night (1996)
100317   Big Night (1996)
73051    Big Night (1996)
142315   Big Night (1996)
70388    Big Night (1996)
70684    John Dies at the End (2012)
81020    Last Man Standing (1996)
7427     Last Man Standing (1996)
64695    Last Man Standing (1996)
20219    Last Man Standing (1996)
8802     It's Such a Beautiful Day (2012)
10817    It's Such a Beautiful Day (2012)

```

3ème étape : Compter le nombre d'utilisateurs par film

Objectif

Compter combien d'utilisateurs ont aimé chaque film.

Mise en œuvre

Mapper : SKCountUsersPerMovieMapper

- Transforme `(userId, movieTitle)` en `(movieTitle, 1)`.

Reducer : SKCountUsersPerMovieReducer

- Compte les occurrences de chaque film.

Driver : SKCountUsersPerMovieDriver

- Configure le job Hadoop.

Commandes

1. Exécuter le job :

```
hadoop jar /tmp/hadoop-first-code-1.0-SNAPSHOT.jar org.hadoop.examples.hadoop.project.driver.SKCountUsersPerMovieDriver /output_movies_user_likes /output_users_per_movie
```

2. Résultats :

```
hdfs dfs -cat /output_users_per_movie/part-*
```

Résultat (extrait)

Zero Effect (1998)	4
Zootopia (2016)	29

```
Youth in Revolt (2009) 1
Z (1969) 2
Zabriskie Point (1970) 1
Zach Galifianakis: Live at the Purple Onion (2006) 1
Zack and Miri Make a Porno (2008) 3
Zazie dans le métro (1960) 1
Zeitgeist: Addendum (2008) 1
Zeitgeist: Moving Forward (2011) 1
Zeitgeist: The Movie (2007) 3
Zero Dark Thirty (2012) 4
Zero Effect (1998) 4
Zodiac (2007) 6
Zombieland (2009) 16
```

```
root@0676b262fa85:/# hdfs dfs -cat /output_users_per_movie/part-*
2025-01-19 04:15:22,840 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localhostTrusted = false, remoteHostTrusted = false
""Great Performances"" Cats (1998)" 1
""burbs 6
"10 1
"13th Warrior 10
"2 Fast 2 Furious (Fast and the Furious 2 9
"20 17
"36th Chamber of Shaolin 1
"39 Steps 9
"4 Months 1
```

4ème étape : Grouper les films par nombre d'utilisateurs

Objectif

Grouper les films par le nombre d'utilisateurs qui les ont aimés.

Mise en œuvre

Mapper : SKInvertKeyValueMapper

- Transforme `(movieTitle, userCount)` en `(userCount, movieTitle)`.

Reducer : SKGroupMoviesByUserCountReducer

- Regroupe les films par `userCount`.

Driver : SKGroupMoviesDriver

- Configure le job Hadoop.

Commandes

1. Exécuter le job :

```
hadoop jar /tmp/hadoop-first-code-1.0-SNAPSHOT.jar org.hadoop.examples.hadoop.project.driver.SKGroupMoviesDriver /output_users_per_movie /output_grouped_movies
```

2. Résultats :

```
hdfs dfs -cat /output_grouped_movies/part-*
```

Résultat (extrait)

```
4    Zero Effect (1998), Zulu (1964)
29   Zootopia (2016)
```

```
1252  Casino (1995)
1277  "Matrix
1278  Léon: The Professional (a.k.a. The Professional) (Léon) (1994)
1406  "Silence of the Lambs
1436  Leaving Las Vegas (1995)
1855  Apollo 13 (1995)
1901  Babe (1995)
1961  "Godfather
1980  Taxi Driver (1976)
2041  Schindler's List (1993)
2660  Heat (1995)
2672  Sense and Sensibility (1995)
2950  Forrest Gump (1994)
3645  Twelve Monkeys (a.k.a. 12 Monkeys) (1995)
4338  Seven (a.k.a. Se7en) (1995)
4732  Braveheart (1995)
5441  Pulp Fiction (1994)
5915  Star Wars: Episode IV - A New Hope (1977)
6379  "Usual Suspects
7281  Toy Story (1995)
7465  "Shawshank Redemption
root@0676b262fa85:/#
```

Analyse et perspectives

Problèmes rencontrés et solutions

1. Fichiers introuvables dans HDFS :

- Problème corrigé en vérifiant les chemins d'entrée et en remplaçant les fichiers dans HDFS.

2. Structure incorrecte des données :


- Ajustement des délimiteurs et validation des lignes dans les Mappers.

3. Résultats vides :

- Analyse approfondie des Reducers pour détecter les erreurs logiques.

Conclusion

Le projet a permis d’explorer les concepts clés de MapReduce et d’Hadoop à travers des questions avancées, notamment des jointures et des regroupements complexes. Les résultats obtenus répondent avec succès aux objectifs.



All Applications

Log

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Ri
5	0	5	0	0 B	16 GB	0 B	0	8	0	

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes	Shutdown Nod
1	0	0	0	0	0	

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation	Maximum Cluster Application Priority
Capacity Scheduler	[memory-mb (unit=Mi), vcores]	<memory:1024, vCores:1>	<memory:8192, vCores:4>	0

Show 20 entries

ID	User	Name	Application Type	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU VCo	Allocated Memory MB	Reserved CPU VCo	Reserved Memory MB	% of Queue	% of Cluster	Progress	Tracking UI
application_1736826468023_0014	root	Highest Rated Tag	MAPREDUCE	default	0	Tue Jan 14 14:05:14 +0100 2025	Tue Jan 14 14:05:15 +0100 2025	Tue Jan 14 14:07:38 +0100 2025	FINISHED	SUCCEEDED	N/A	N/A	N/A	N/A	N/A	0.0	0.0		History
application_1736826468023_0015	root	Highest Rated Movie By Genre	MAPREDUCE	default	0	Tue Jan 14 14:06:25 +0100 2025	Tue Jan 14 14:07:46 +0100 2025	Tue Jan 14 14:11:19 +0100 2025	FINISHED	FAILED	N/A	N/A	N/A	N/A	N/A	0.0	0.0		History
application_1736826468023_0012	root	Highest Rated Movie By User	MAPREDUCE	default	0	Tue Jan 14 11:54:22 +0100 2025	Tue Jan 14 11:54:23 +0100 2025	Tue Jan 14 11:57:00 +0100 2025	FINISHED	SUCCEEDED	N/A	N/A	N/A	N/A	N/A	0.0	0.0		History
application_1736826468023_0013	root	Highest Rated Movie By User	MAPREDUCE	default	0	Tue Jan 14 13:54:04 +0100 2025	Tue Jan 14 13:54:07 +0100 2025	Tue Jan 14 13:58:12 +0100 2025	FINISHED	SUCCEEDED	N/A	N/A	N/A	N/A	N/A	0.0	0.0		History
application_1736826468023_0010	root	Average Rating By Movie	MAPREDUCE	default	0	Tue Jan 14 10:31:33 +0100 2025	Tue Jan 14 10:32:33 +0100 2025	Tue Jan 14 10:33:23 +0100 2025	FINISHED	FAILED	N/A	N/A	N/A	N/A	N/A	0.0	0.0		History
application_1736826468023_0011	root	Highest Rated Movie	MAPREDUCE	default	0	Tue Jan 14 10:31:59 +0100 2025	Tue Jan 14 10:33:31 +0100 2025	Tue Jan 14 10:36:57 +0100 2025	FINISHED	FAILED	N/A	N/A	N/A	N/A	N/A	0.0	0.0		History