Cours
Mise en œuvre des Framework D'Intelligence Artificielle et Big data

Cycle Ingénieur
INDIA, Semestre 5

Pr. Abderrahim El Qadi
Département Mathématique Appliquée et Génie Informatique
ENSAM, Université Mohammed V de Rabat

A.U. 2023/2024

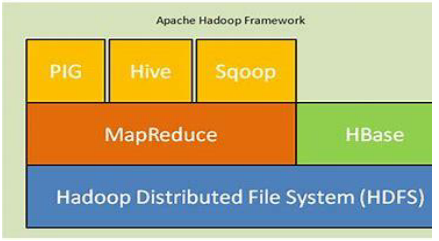
4eme partie
8. HBase
9. Scoop

Références

- Book. Big Data Using HaDoop anD Hive. Nitin Kumar
- Book. B IG DATA ANALYTICS, Introduction to Hadoop, Spark, and Machine-Learning. Raj Kamal.
- Book. Apprenez Sqoop (https://riptutorial.com)
- https://www.guru99.com/hbase-shell-general-commands.html
- https://www.guru99.com/hbase-tutorials.html
- https://www.tutorialspoint.com/hbase/hbase_describe_and_alter.html
- https://learnhbase.wordpress.com/2013/03/02/hbase-shell-commands/
- https://bigdataprogrammers.com/import-csv-data-into-hbase/
- https://sqoop.apache.org/docs/1.4.6/SqoopUserGuide.html

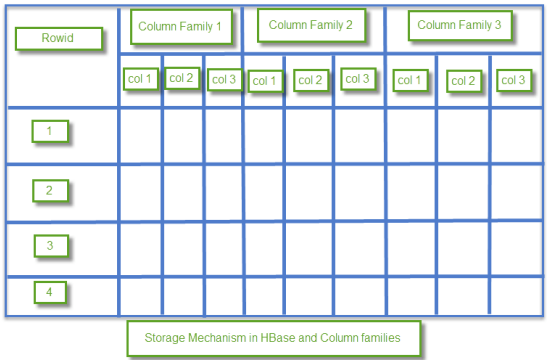
8.

- HBase est une base de données distribuée, non relationnelle et orienté colonnes, développée au-dessus du système de fichier HDFS.
- Elle permet un accès aléatoire en écriture/lecture en temps réel à un très grand ensemble de données.

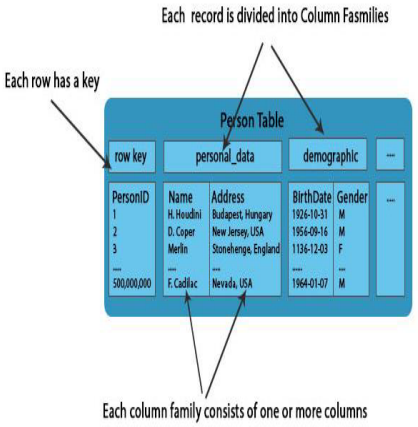


8.1. Modèle de données

- Le modèle se base sur les concepts : Table, Row, Column Family (cf), Column qualifier (cq), Cell.



Mécanisme de stockage HDFS



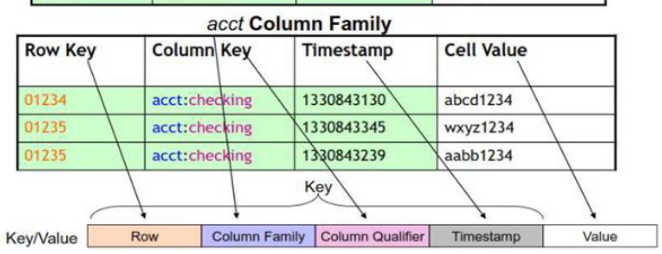
- Les données sont stockées dans une table.
- Chaque ligne a une clé
- Colonne : Il s'agit d'un ensemble de données appartenant à une famille de colonnes, qui est incluse à l'intérieur de la ligne
- Famille de colonnes : Chaque famille de colonnes se compose d'une ou plusieurs colonnes.
- Chaque table contient une collection de familles de colonnes. Ces colonnes ne font pas partie du schéma.

info **Column Family**

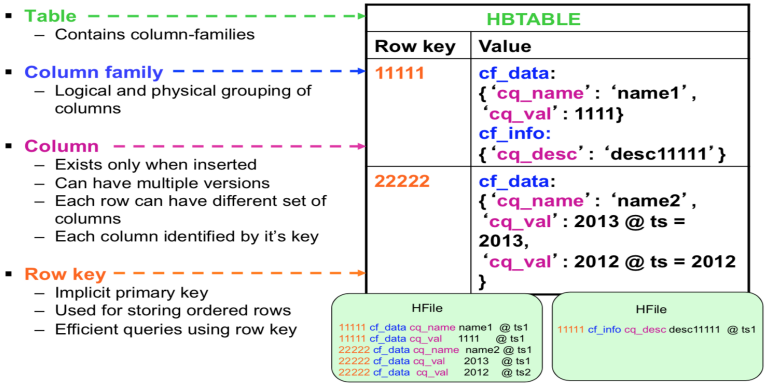
Row Key	Column Key	Timestamp	Cell Value
01234	info:fname	1330843130	John
01234	info:lname	1330843130	Smith
01235	info:fname	1330843345	Michael
01235	info:lname	1330843345	Johnson
01236	info:lname		Mules

acct **Column Family**

Row Key	Column Key	Timestamp	Cell Value
01234	acct:checking	1330843130	abcd1234
01235	acct:checking	1330843345	wxyz1234
01235	acct:checking	1330843239	aabb1234



- Exemple



- **Différences entre RDBMS et HBase.**

	SGBDR	HBase
Définition	RDBMS signifie Relational DataBase Management System (système de gestion de base de données relationnelle).	HBase n'a pas de forme complète.
SQL	Les SGBDR nécessitent le langage SQL, Structured Query Language.	HBase n'a pas besoin de SQL.
Schéma	Le SGBDR a un schéma fixe.	HBase n'a pas de schéma fixe.
Orientation	Le SGBDR est orienté vers les lignes.	HBase est orienté colonnes.
Évolutivité	Les SGBDR sont confrontés à des problèmes d'extensibilité.	HBase est hautement évolutive.
Nature	Le SGBD est statique par nature.	HBase est dynamique par nature.
Récupération des données	La récupération des données dans un SGBDR est lente.	La récupération des données HBase est rapide.
REGLE	Les SGBDR suivent la règle ACID (Atomicité, Cohérence, Isolation et Durabilité).	HBase suit la règle CAP (Consistency, Availability, Partition-tolerance).
Structure des données	Le SGBDR traite les données structurelles.	HBase traite des données structurelles, non structurelles et semi-structurelles.

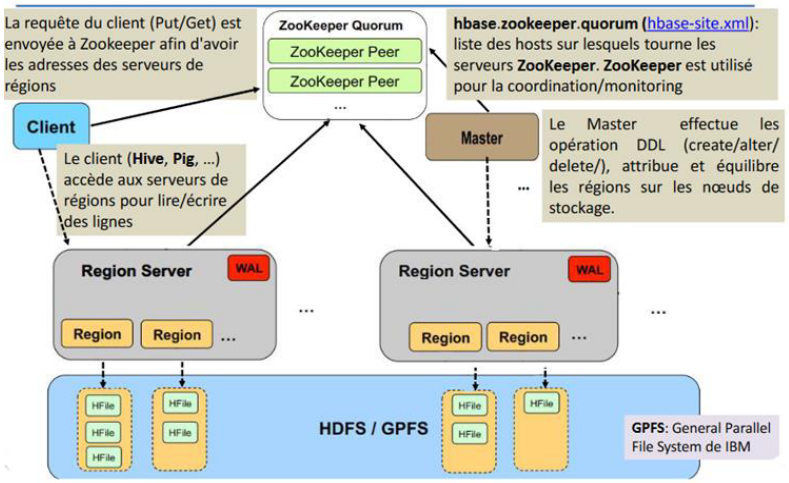
RDBMS

SSN - primary key	Last Name	First Name	Account Number	Type of Account	Timestamp
01234	Smith	John	abcd1234	Checking	20120618...
01235	Johnson	Michael	wxyz1234	Checking	20121118...
01235	Johnson	Michael	aabb1234	Checking	20151123...
01236	Mules	null	null	null	null

HBase

Row key	Value (CF, Column, Version, Cell)
01234	info: {'lastName': 'Smith', 'firstName': 'John'} acct: {'checking': 'abcd1234'}
01235	info: {'lastName': 'Johnson', 'firstName': 'Michael'} acct: {'checking': 'wxyz1234'@ts=2013, 'checking': 'aabb1234'@ts=2012}
01236	info: {'lastName': 'Mules'}

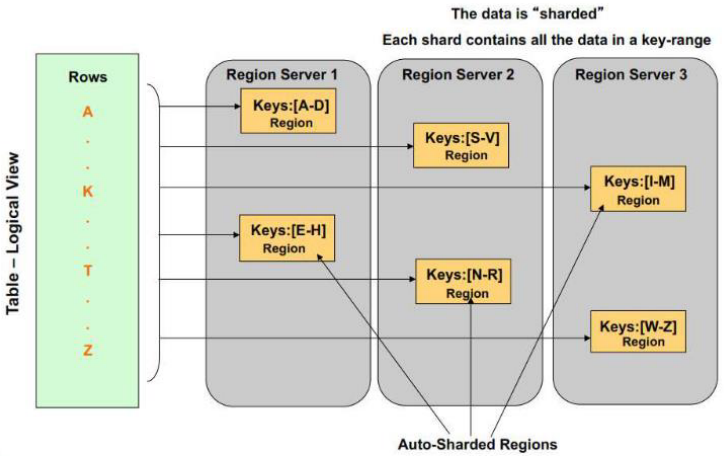
8.2. Architecture HBase



- HBase est composé de trois types de serveurs de type Master/Slave.
  - **Region Servers** : permettent de fournir les données pour lectures et écritures. Pour accéder aux données, les clients communiquent avec les **Region Servers** directement.
  - **HBase HMaster** : gère l'affectation des régions, les opérations de création et suppression de tables.
  - HBase utilise **Zookeeper** comme service de coordination pour maintenir l'état du serveur dans le cluster. Zookeeper sait quels serveurs sont actifs et disponibles, et fournit une notification en cas d'échec d'un serveur.
- Le DataNode de Hadoop permet de stocker les données gérées par Region Server.
  - Toutes les données de HBase sont stockées dans des fichiers HDFS.
  - Les Region Servers sont colocalisés avec les DataNodes.
- Le NameNode permet de maintenir les métadonnées sur tous les blocs physiques qui forment les fichiers.

- **HBase : Region**
  - Une région est une partition horizontale d'une table avec une ligne de début et une ligne de fin. (taille par défaut: 256M)
  - Les régions sont l'élément de base de la disponibilité et de la distribution des tables.
  - Une région est automatiquement divisée par le serveur de la région lorsqu'elle dépasse une taille spécifiée.
  - Périodiquement, un équilibreur de charge déplace les régions dans le cluster pour équilibrer la charge.
  - Lorsqu'un serveur de région est défaillant, ses régions seront réaffectées à d'autres serveurs de régions.

- Stockage des tables



8.3. Les commandes de base de HBase

– Commandes générales de l'interpréteur de commandes HBase

Nom utilisateur	hbase>whoami
Afficher l'état du cluster.	hbase>status
Version	hbase>version

– Commandes LDD :

Création table	create '<table_name>', '<column_family_name>' ou create '<table_name>', {'NAME=><column_family_name>'}
Création la table t1 avec 3 familles de colonnes	create 't1', {NAME => 'f1'}, {NAME => 'f2'}, {NAME => 'f3'}
La liste des tables dans HBase	list
Informations détaillées sur la table	describe '<table_name>'
Désactiver une table	disable '<table_name>'
Désactiver toutes les tables correspondant à l'expression régulière donnée	disable_all 't.*'
Active la table	enable '<table_name>'
Affichage tous les filtres présents	show_filters
Suppression d'une table	drop '<table_name>'
Suppression la famille de colonnes 'f1' dans la table 't1',	alter 't1', 'delete' => 'f1'

Désactiver toutes les tables correspondant à l'expression régulière donnée	drop_all 't.*'
Modification du schéma de la famille de colonnes	alter <table_name>, NAME=><column_family_name>, VERSIONS=>5
Modification de plusieurs familles de colonnes	hbase> alter 't1', 'f1', {NAME => 'f2', IN_MEMORY => true}, {NAME => 'f3', VERSIONS => 5}

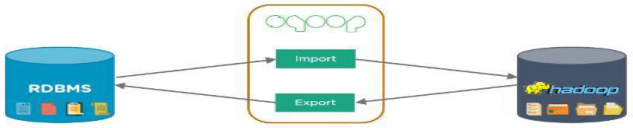
– Commandes LMD

Compte nombre de lignes	count '<table_name>'
vérifier si la valeur entrée/insérée est correctement présente dans le tableau ou non.	scan '<table_name>'
ajouter/mettre/insérer les données dans une table spécifique.	put '<table_name>', '<row_key>', '<col>', '<data>'
voir le contenu de la table	get '<table_name>', '<row_key>','<col_name>'
supprimer les données réelles	delete '<table_name>', '<row>', '<col>'
supprimer toutes les cellules	deleteall '<table_name>', '<row>', '<col>'
conserver le schéma de la table	truncate '<table_name>'
Utilisation du filtre SingleColumnValueFilter	SingleColumnValueFilter ('<family>', '<qualifier>', <compare operator>, '<comparator>')

9. Sqoop

APACHE SQOOP

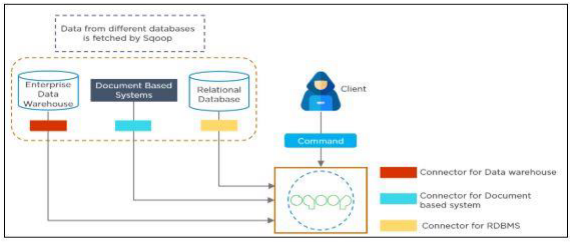
– Apache Sqoop est un outil largement utilisé pour transférer de grandes quantités de données de Hadoop vers les serveurs de bases de données relationnelles et vice-versa.



– Sqoop possède plusieurs fonctionnalités, ce qui le rend utile dans le monde du Big Data :

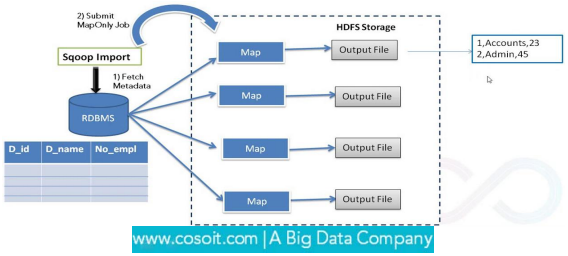
1. Importation/exportation parallèle : Sqoop utilise le framework YARN pour importer et exporter des données. Cela permet d'obtenir une tolérance aux pannes en plus du parallélisme.
2. Importation les résultats d'une requête SQL dans HDFS.
3. Connecteurs pour toutes les principales bases de données SGBDR
4. Fournit une charge complète et incrémentielle

10.1. Sqoop Architecture



1. Le client soumet la commande d'import/export pour importer ou exporter des données.
2. Sqoop récupère les données de différentes bases de données.
3. Plusieurs mappeurs effectuent des tâches de mappage pour charger les données sur HDFS.
4. De même, de nombreuses tâches de mappage exportent les données de HDFS vers le SGBDR à l'aide de la commande d'exportation Sqoop.

10.2. Importation Sqoop



1. Dans cet exemple, les données d'une entreprise sont présentes dans le SGBDR. Toutes ces métadonnées sont envoyées à l'import Sqoop. Scoop effectue ensuite une introspection de la base de données pour recueillir des métadonnées (informations de clé primaire).
2. Il soumet ensuite une tâche de mappage uniquement.
3. Sqoop divise le jeu de données d'entrée en fractions et utilise des tâches de map pour envoyer les fractions vers HDFS.

Commandes de contrôle Sqoop pour importer des données SGBDR

- **Append:** Append data to an existing dataset in HDFS. –append
- **Columns:** columns to import from the table. –columns <col,col……>
- **Where:** where clause to use during import. –where <where clause>

– Sqoop peut importer :
  - Une partie d'une table.
  - Une table complète.
  - Plusieurs tables complètes.

- Sqoop import
  - Importing a Table
    - Sqoop tool 'import' is used to import table data from the table to the Hadoop file system as a text file or a binary file.

```
$ sqoop import --connect jdbc:mysql://localhost/userdb --username root --table emp --m 1
```

  - Importing into a directory
    - We can specify the target directory while importing table data into HDFS using the Sqoop import tool.

```
$ sqoop import --connect jdbc:mysql://localhost/userdb --username root --table emp_add --m 1 --target-dir /queryresult
```

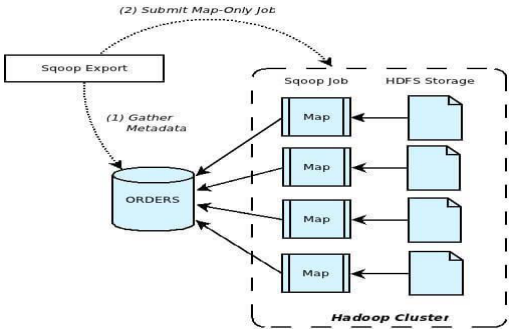

- Sqoop import
  - Import subset of table data
    - We can import a subset of a table using the 'where' clause in Sqoop import tool.

```
$ sqoop import \ --connect jdbc:mysql://localhost/userdb --username root --table emp_add --m 1
--where "city ='Rabat'" --target-dir /wherequery
```

  - Incremental Import
    - Incremental import is a technique that imports only the newly added rows in a table. It is required to add 'incremental', 'check-column', and 'last-value' options to perform the incremental import.

```
$ sqoop import --connect jdbc:mysql://localhost/userdb --username root --table emp --m 1
--incremental append --check-column id -last value 1205
```

10.3. Exportation Sqoop



1. La première étape consiste à recueillir les métadonnées par le biais de l'introspection.
2. Sqoop divise ensuite le jeu de données d'entrée en fractions et utilise des tâches de map pour envoyer les fractionnements vers le SGBDR.

- Sqoop import-all-tables
  - Importing all tables
    - The following command is used to import all the tables from the **userdb** database.

```
$ sqoop import-all-tables --connect jdbc:mysql://localhost/userdb --username root
```

- Sqoop export
  - Create database in mysql
    - It exports the database from HDFS to MYSQL.

```
$ sqoop export \
--connect jdbc:mysql://localhost/db \
--username root \
--table employee \
--export-dir /emp/emp_data
```


- Sqoop Jobs
  - Create Job
    - The following command is used to create a job that is importing data from the **employee** table in the **db** database to the HDFS file.

```
$ sqoop job --create myjob --import --connect jdbc:mysql://localhost/db --username root --table employee --m 1
```

  - Verify Job
    - **'--list'** argument is used to verify the saved jobs.

```
$ sqoop job --list
```

  - Inspect Job
    - **'--show'** argument is used to inspect or verify particular jobs

```
$ sqoop job --show myjob
```

  - Execute Job
    - **'--exec'** option is used to execute a saved job.

```
$ sqoop job --exec myjob
```


- Eval
  - The eval tool allows users to execute user-defined queries against respective database servers and preview the result in the console.
  - Select

```
$ sqoop eval --connect jdbc:mysql://localhost/db --username root --query "SELECT * FROM employee LIMIT 3"
```

  - Insert

```
$ sqoop eval --connect jdbc:mysql://localhost/db --username root -e "INSERT INTO employee VALUES(1207,'James','UI dev',15000,'TP')"
```

- List databases
  - This command is used to list all the databases in the MySQL database server.

```
$ sqoop list-databases --connect jdbc:mysql://localhost/ --username root
```

- List Tables
  - This command is used to list all the tables in the **userdb** database of MySQL database server.

```
$ sqoop list-tables --connect jdbc:mysql://localhost/userdb --username root
```