



Pr. Ghazouani mohamed

1

9 décembre 2024

Les bases de données relationnelles

Pr. M. Ghazouani

À ce stade, une nouvelle solution est nécessaire pour accéder à n'importe quel point de données en une seule unité de temps (accès aléatoire).

Les bases telles que HBase, Cassandra et MongoDB sont des bases de données qui stockent d'énormes quantités de données et accèdent aux données de manière aléatoire.

2

1

Les bases de **données relationnelles** sont basées sur le modèle **relationnel**, un moyen intuitif et simple de représenter des **données** dans des tables. Dans une **base de données relationnelle**, chaque ligne de la table est un enregistrement avec un identifiant unique appelé clé.

Les avantages de BDR: 1- plusieurs requêtes de lecture au même temps 2- cohérence

Désavantages : ne peut pas être scale horizontalement, ne peut pas être mis en place d'une façon distribuée.

HDFS manque la capacité de lecture/écriture aléatoire. Il convient aux accès séquentiels aux données.

```

    graph TD
      panier[panier]
      produit[produit]
      panier_elem[panier_elem]

      panier --> panier_elem
      produit --> panier_elem
  
```

panier		
id	uuid	PK
nom	varchar	

produit		
id	uuid	PK
libelle	varchar	
prix	number(10,2)	

panier_elem		
panier_id	uuid	PK1
product_id	uuid	PK2
quantite	int	

Qu'est-ce que HBase?

- **HBase** est un système de gestion de base de données **non-relationnelles** distribué, écrit en Java, disposant d'un stockage structuré pour les grandes tables.
- **HBase** est inspirée des publications de Google sur BigTable. Comme BigTable, c'est une base de données orientée colonnes. Basées sur une architecture maître/esclave, les bases de données de ce type sont capables de gérer d'énormes quantités d'informations (plusieurs milliards de lignes par table)
- **HBase** est un sous-projet d'Hadoop. La base de données HBase s'installe généralement sur le système de fichiers HDFS d'Hadoop pour faciliter la distribution, même si ce n'est pas obligatoire.

Mécanisme de stockage dans HBase

Indexation

Key range

Id 1
....
Id 100

panier		
id	uuid	PK
nom	varchar	

produit		
id	uuid	PK
libelle	varchar	
prix	number(10,2)	

panier_elem		
panier_id	uuid	PK1
product_id	uuid	PK2
quantite	int	



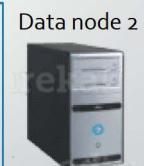
Key range

Id 101
....
Id 200

panier		
id	uuid	PK
nom	varchar	

produit		
id	uuid	PK
libelle	varchar	
prix	number(10,2)	

panier_id	uuid	PK1
product_id	uuid	PK2
quantite	int	



Key range

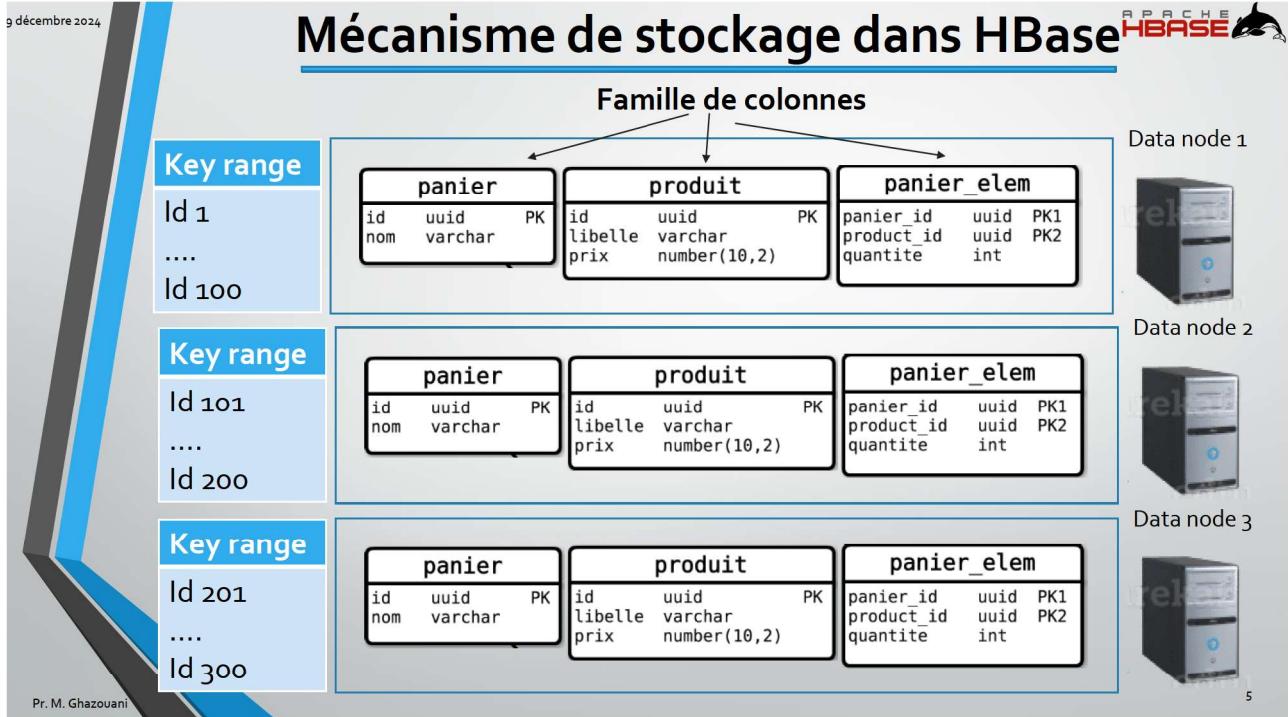
Id 201
....
Id 300

panier		
id	uuid	PK
nom	varchar	

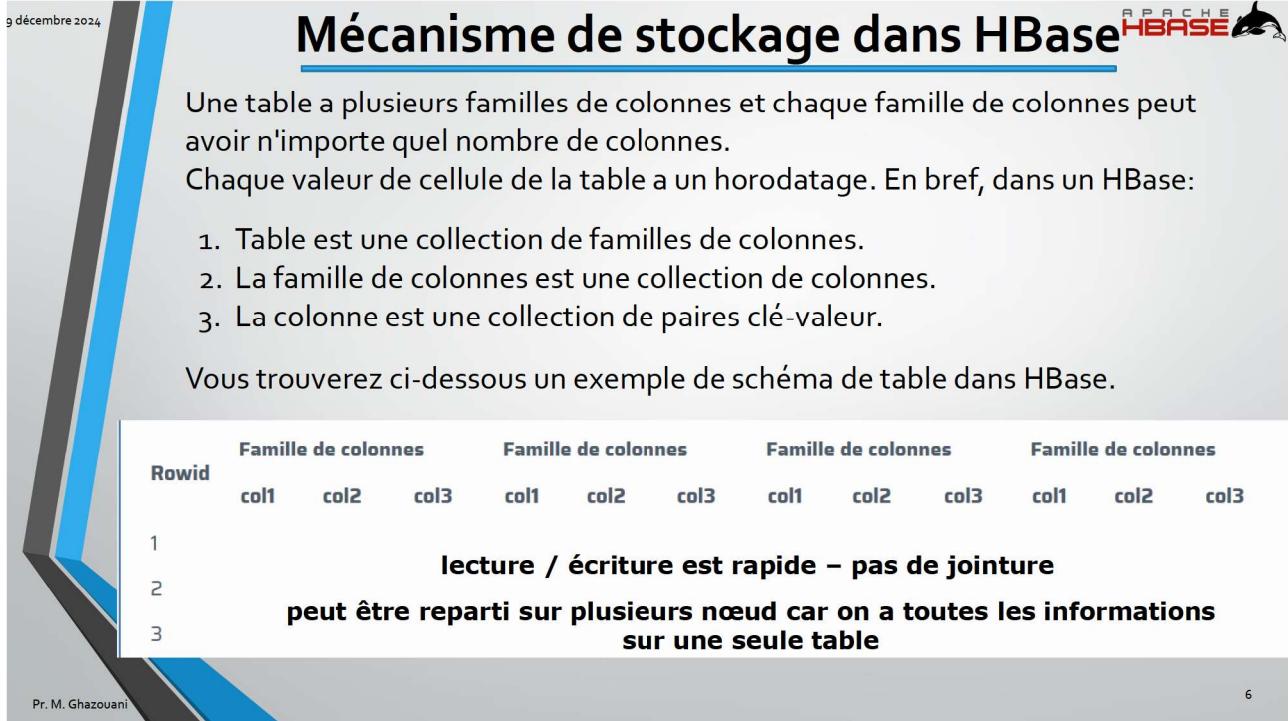
produit		
id	uuid	PK
libelle	varchar	
prix	number(10,2)	

panier_id	uuid	PK1
product_id	uuid	PK2
quantite	int	





5



6

9 décembre 2024

Mécanisme de stockage dans HBase



Le concept de base de données en HBase est la **table HBase**. Une table HBase est un tableau multidimensionnel de données distribué et persisté sur le HDFS sous forme de fichiers spécifiques appelés **HFiles**. Voici les dimensions qui forment ce tableau multidimensionnel :

	Row Key	Prénom	Personnel	Age	téléphone	Contact	ville
00001	Juvénal timestamp: 10/09/2011 13:05:17:09	22 timestamp: 10/09/2011 13:05:17:09		06 90 98 76 52 timestamp: 10/09/2011 13:05:17:09	Douala timestamp: 10/09/2011 13:05:17:09		
		25 timestamp: 20/09/2013 15:00:05:09			Lille timestamp: 20/09/2013 15:00:05:09		
					Paris timestamp: 30/10/2016 16:18:50:10		
00002	Paul timestamp: 10/09/2011 13:10:05:09	30 timestamp: 10/09/2011 13:10:05:09		07 90 94 86 52 timestamp: 10/09/2011 13:10:05:09	Nancy timestamp: 10/09/2011 13:10:05:09		
00003	Jean timestamp: 12/09/2011 11:30:20:09	34 timestamp: 12/09/2011 11:30:20:09		06 74 98 76 25 timestamp: 12/09/2011 11:30:20:09	Marseille timestamp: 12/09/2011 11:30:20:09		

Pr. M. Ghazouani

7

9 décembre 2024

Mécanisme de stockage dans HBase



- La première dimension c'est la "**row key**" (clé de ligne).
- Chaque ligne du tableau HBase est identifiée de façon unique par une clé "row key".
- La row key n'est pas nécessairement une colonne parmi l'ensemble des colonnes désignées comme telles par le développeur, c'est une colonne interne à la structure de la table HBase, similaire à la colonne de numéros qui identifient chaque ligne d'une feuille de calcul Excel.

	A	B	C
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			

	Personnel		Contact	
Row Key	Prénom	Age	téléphone	ville
00001	Juvénal timestamp: 10/09/2011 13:05:17:09	22 timestamp: 10/09/2011 13:05:17:09	06 90 98 76 52 timestamp: 10/09/2011 13:05:17:09	Douala timestamp: 10/09/2011 13:05:17:09
		25 timestamp: 20/09/2013 15:00:05:09		Lille timestamp: 20/09/2013 15:00:05:09
				Paris timestamp: 30/10/2016 16:18:50:10
00002	Paul timestamp: 10/09/2011 13:10:05:09	30 timestamp: 10/09/2011 13:10:05:09	07 90 94 86 52 timestamp: 10/09/2011 13:10:05:09	Nancy timestamp: 10/09/2011 13:10:05:09
00003	Jean timestamp: 12/09/2011 11:30:20:09	34 timestamp: 12/09/2011 11:30:20:09	06 74 98 76 25 timestamp: 12/09/2011 11:30:20:09	Marseille timestamp: 12/09/2011 11:30:20:09

Pr. M. Ghazouani

8

9 décembre 2024

Mécanisme de stockage dans HBase



- La deuxième dimension est la **famille de colonne** (column family), d'où HBase tire son nom de SGBD orienté colonne.
- Chaque famille de colonnes est persistée dans un HFile séparé, et chaque HFile possède son propre jeu de paramètres de configuration.
- Cette dimension est très importante, car elle définit la façon dont les données sont persistées et accédées.
- À cet effet, lors de la modélisation de la table HBase, vous veillerez particulièrement à placer les colonnes les plus utilisées dans la même famille de colonnes.

clé: 1001	produit: livre Hadoop pour les consultants	Qtité: 7	Prix: 13 euros
clé: 1001	produit: livre Hadoop pour les consultants 2		Prix: 14 euros
clé: 1001		Qtité: 1	Prix: 19 euros
clé: 1002	produit: véhicule citadine	Qtité: 1	Prix: 13300 euros
clé: 1002	produit: véhicule citadine avec démarrage automatique		Prix: 13300 euros
clé: 1002		Qtité: 2	Prix: 16000 euros
clé: 1002	produit: costume 3 pièces marque LAMBROZINI		Prix: 1000 euros

Row Key	Prénom	Personnel		Contact	
		Age		téléphone	ville
00001	Juvénal	22		06 90 98 76 52	Douala
	<i>timestamp:</i> 10/09/2011 13:05:17:09	<i>timestamp:</i> 10/09/2011 13:05:17:09		<i>timestamp:</i> 10/09/2011 13:05:17:09	<i>timestamp:</i> 10/09/2011 13:05:17:09
00002		25			Lille
	<i>timestamp:</i> 20/09/2013 15:00:05:09	<i>timestamp:</i> 20/09/2013 15:00:05:09		<i>timestamp:</i> 20/09/2013 15:00:05:09	<i>timestamp:</i> 20/09/2013 15:00:05:09
00003	Paul	30		07 90 94 86 52	Nancy
	<i>timestamp:</i> 10/09/2011 13:10:05:09	<i>timestamp:</i> 10/09/2011 13:10:05:09		<i>timestamp:</i> 10/09/2011 13:10:05:09	<i>timestamp:</i> 10/09/2011 13:10:05:09
	Jean	34		06 74 98 76 25	Marseille
	<i>timestamp:</i> 12/09/2011 11:30:20:09	<i>timestamp:</i> 12/09/2011 11:30:20:09		<i>timestamp:</i> 12/09/2011 11:30:20:09	<i>timestamp:</i> 12/09/2011 11:30:20:09

9

9 décembre 2024

Mécanisme de stockage dans HBase



- À la différence des colonnes, les familles de colonnes elles, sont statiques, c'est-à-dire définies à la création de la table et fixes tout au long de leur vie.
- les colonnes sont dynamiques, c'est-à-dire ne sont pas définies lors de la création de la table, mais sont dynamiquement créées lors des opérations d'écriture/mise à jour de la donnée dans la colonne de la ligne.
- HBase traite les colonnes comme une table dynamique de paires de clé/valeur. Comme résultat, chaque ligne/famille de colonnes peut contenir des jeux arbitraires de colonnes.

clé: 1001	produit: livre Hadoop pour les consultants	Qtité: 7	Prix: 13 euros
clé: 1001	produit: livre Hadoop pour les consultants 2		Prix: 14 euros
clé: 1001		Qtité: 1	Prix: 19 euros
clé: 1002	produit: véhicule citadine	Qtité: 1	Prix: 13300 euros
clé: 1002	produit: véhicule citadine avec démarrage automatique		Prix: 13300 euros
clé: 1002		Qtité: 2	Prix: 16000 euros
clé: 1002	produit: costume 3 pièces marque LAMBROZINI		Prix: 1000 euros

Row Key	Prénom	Personnel		Contact	
		Age		téléphone	ville
00001	Juvénal	22		06 90 98 76 52	Douala
	<i>timestamp:</i> 10/09/2011 13:05:17:09	<i>timestamp:</i> 10/09/2011 13:05:17:09		<i>timestamp:</i> 10/09/2011 13:05:17:09	<i>timestamp:</i> 10/09/2011 13:05:17:09
00002		25			Lille
	<i>timestamp:</i> 20/09/2013 15:00:05:09	<i>timestamp:</i> 20/09/2013 15:00:05:09		<i>timestamp:</i> 20/09/2013 15:00:05:09	<i>timestamp:</i> 20/09/2013 15:00:05:09
00003	Paul	30		07 90 94 86 52	Nancy
	<i>timestamp:</i> 10/09/2011 13:10:05:09	<i>timestamp:</i> 10/09/2011 13:10:05:09		<i>timestamp:</i> 10/09/2011 13:10:05:09	<i>timestamp:</i> 10/09/2011 13:10:05:09
	Jean	34		06 74 98 76 25	Marseille
	<i>timestamp:</i> 12/09/2011 11:30:20:09	<i>timestamp:</i> 12/09/2011 11:30:20:09		<i>timestamp:</i> 12/09/2011 11:30:20:09	<i>timestamp:</i> 12/09/2011 11:30:20:09

10

9 décembre 2024

Mécanisme de stockage dans HBase



- La troisième dimension est la **colonne** (column qualifier). Une colonne c'est l'adresse d'une série de données dans une famille de colonnes.
- Les colonnes sont dynamiques et peuvent être différentes d'une ligne à une autre, en d'autres termes, des valeurs peuvent être manquantes pour la ligne d'une colonne (NULL dans les bases de données traditionnelles), auquel cas la colonne n'est pas créée pour la ligne en question.

clé: 1001	produit: livre Hadoop pour les consultants	Qtité: 7	Prix: 13 euros
clé: 1001	produit: livre Hadoop pour les consultants 2		Prix: 14 euros
clé: 1001		Qtité: 1	Prix: 19 euros
clé: 1002	produit: véhicule citadine	Qtité: 1	Prix: 13300 euros
clé: 1002	produit: véhicule citadine avec démarrage automatique		Prix: 16000 euros
clé: 1002		Qtité: 2	Prix: 1000 euros
clé: 1002	produit: costume 3 pièces marque LAMBROZINI		Prix: 1000 euros

Row Key	Personnel		Contact	
	Prénom	Age	téléphone	ville
00001	Juvénal timestamp: 10/09/2011 13:05:17:09	22 timestamp: 10/09/2011 13:05:17:09	06 90 98 76 52 timestamp: 10/09/2011 13:05:17:09	Douala timestamp: 10/09/2011 13:05:17:09
		25 timestamp: 20/09/2013 15:00:05:09		Lille timestamp: 20/09/2013 15:00:05:09
00002	Paul timestamp: 10/09/2011 13:10:05:09	30 timestamp: 10/09/2011 13:10:05:09	07 90 94 86 52 timestamp: 10/09/2011 13:10:05:09	Nancy timestamp: 10/09/2011 13:10:05:09
00003	Jean timestamp: 12/09/2011 11:30:20:09	34 timestamp: 12/09/2011 11:30:20:09	06 74 98 76 25 timestamp: 12/09/2011 11:30:20:09	Marseille timestamp: 12/09/2011 11:30:20:09

11

9 décembre 2024

Mécanisme de stockage dans HBase



- La quatrième dimension c'est la **cellule** (cell) ou **valeur** (pour faire référence au contenu de la cellule). Une cellule HBase est l'intersection entre une row key, une famille de colonnes et une colonne. Les données sont stockées dans les cellules.
- À la différence des cellules dans le sens Excel du terme, les cellules HBase ne stockent pas uniquement les données textuelles ou des nombres. Vous pouvez stocker dans chaque cellule un PDF, une vidéo, une image, un texte et n'importe quel type de données numériques. Elles y sont stockées sous forme de type générique Byte[], ainsi, vous n'avez pas à typer les colonnes en HBase.

Row Key	Personnel		Contact	
	Prénom	Age	téléphone	ville
00001	Juvénal timestamp: 10/09/2011 13:05:17:09	22 timestamp: 10/09/2011 13:05:17:09	06 90 98 76 52 timestamp: 10/09/2011 13:05:17:09	Douala timestamp: 10/09/2011 13:05:17:09
		25 timestamp: 20/09/2013 15:00:05:09		Lille timestamp: 20/09/2013 15:00:05:09
00002	Paul timestamp: 10/09/2011 13:10:05:09	30 timestamp: 10/09/2011 13:10:05:09	07 90 94 86 52 timestamp: 10/09/2011 13:10:05:09	Nancy timestamp: 10/09/2011 13:10:05:09
00003	Jean timestamp: 12/09/2011 11:30:20:09	34 timestamp: 12/09/2011 11:30:20:09	06 74 98 76 25 timestamp: 12/09/2011 11:30:20:09	Marseille timestamp: 12/09/2011 11:30:20:09

Pr. M. Ghazouani

12

9 décembre 2024

Mécanisme de stockage dans HBase



- La dernière dimension est le **TimeStamp**. Il faut savoir qu'HBase ne fait pas de différence entre les ajouts de lignes dans la table (opération d'écriture) et leur mise à jour (opération de modification d'une ligne). Tout comme dans le HDFS et dans un Data Warehouse, chaque opération de mise à jour est un ajout d'une nouvelle version de la même ligne de données.
- La distinction de versions d'une même ligne se fait à l'aide d'une valeur horodatée appelée TimeStamp. Ce TimeStamp c'est la date et l'heure, à la milliseconde près, auquelles la mise à jour a été effectuée.
- Par défaut, HBase stocke les trois dernières versions d'une valeur de colonne (tout en supprimant automatiquement les anciennes versions). La profondeur du "Versioning" peut être contrôlée lors de la création de la table.

Pr. M. Ghazouani

Personnel				Contact
Row Key	Prénom	Age	téléphone	
00001	Juvénal timestamp: 10/09/2011 13:05:17:09	22 timestamp: 10/09/2011 13:05:17:09	06 90 98 76 52 timestamp: 10/09/2011 13:05:17:09	Douala timestamp: 10/09/2011 13:05:17:09
		25 timestamp: 20/09/2013 15:00:05:09		Lille timestamp: 20/09/2013 15:00:05:09
				Paris timestamp: 30/10/2016 16:18:50:10
00002	Paul timestamp: 10/09/2011 13:10:05:09	30 timestamp: 10/09/2011 13:10:05:09	07 90 94 86 52 timestamp: 10/09/2011 13:10:05:09	Nancy timestamp: 10/09/2011 13:10:05:09
00003	Jean timestamp: 12/09/2011 11:30:20:09	34 timestamp: 12/09/2011 11:30:20:09	06 74 98 76 25 timestamp: 12/09/2011 11:30:20:09	Marseille timestamp: 12/09/2011 11:30:20:09

13

9 décembre 2024

Mécanisme de stockage dans HBase



La représentation de la table HBase de notre exemple est en paires de clés/valeurs. L'horodatage est propre à chaque cellule, pas globale à la ligne. En d'autres termes, chaque cellule est horodatée du TimeStamp du moment auquel elle a été créée. La row key n'identifie pas les lignes, mais la combinaison « famille, colonne, cellule, TimeStamp ».

Les cellules sont les intersections entre « row key, TimeStamp, famille de colonnes et colonne ». Par exemple, la cellule à l'intersection « 00003, 12/09/2011 11:30:20:09, Contact, ville » contient la valeur ?

Personnel				Contact
Row Key	Prénom	Age	téléphone	ville
00001	Juvénal timestamp: 10/09/2011 13:05:17:09	22 timestamp: 10/09/2011 13:05:17:09	06 90 98 76 52 timestamp: 10/09/2011 13:05:17:09	Douala timestamp: 10/09/2011 13:05:17:09
		25 timestamp: 20/09/2013 15:00:05:09		Lille timestamp: 20/09/2013 15:00:05:09
				Paris timestamp: 30/10/2016 16:18:50:10
00002	Paul timestamp: 10/09/2011 13:10:05:09	30 timestamp: 10/09/2011 13:10:05:09	07 90 94 86 52 timestamp: 10/09/2011 13:10:05:09	Nancy timestamp: 10/09/2011 13:10:05:09
00003	Jean timestamp: 12/09/2011 11:30:20:09	34 timestamp: 12/09/2011 11:30:20:09	06 74 98 76 25 timestamp: 12/09/2011 11:30:20:09	Marseille timestamp: 12/09/2011 11:30:20:09

14

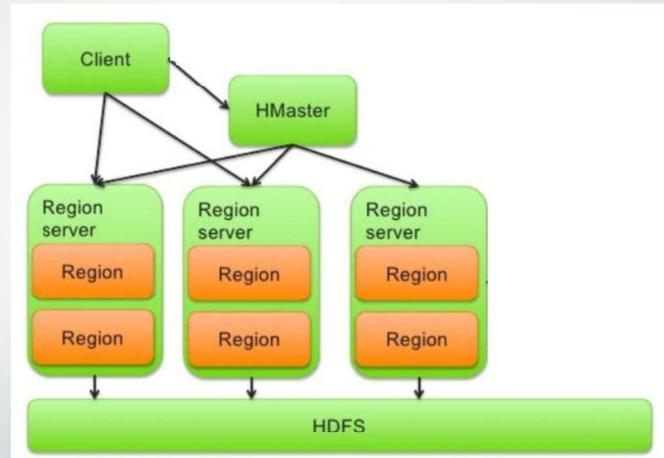
9 décembre 2024

Architecture et fonctionnement du Hbase



- La figure ci-après illustre l'architecture d'un cluster HBase.

Architecture d'un cluster HBase. Le HBase utilise le HDFS pour stocker les tables. Le HDFS voit HBase comme un client à qui il fournit du support de stockage des HFiles qui constituent les tables en blocs.



Pr. M. Ghazouani

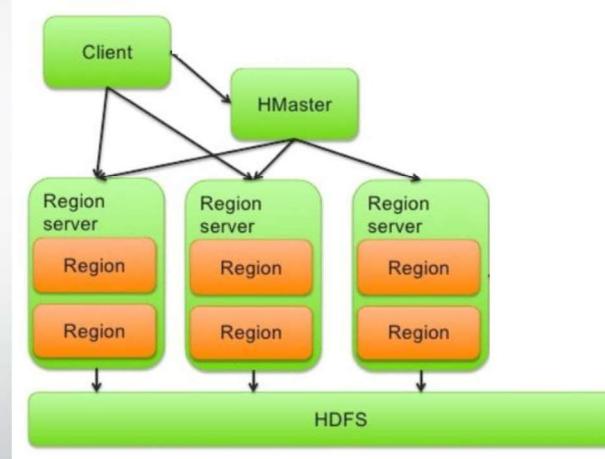
15

9 décembre 2024

Architecture et fonctionnement du HBase



- HBase** est un système de gestion de base de données non-relationalles distribué et en tant que tel, il s'installe sur un cluster d'ordinateurs.
- Comme Hadoop, HBase s'installe sur un cluster en architecture Maître/Eslave. Dans la terminologie HBase, le nœud maître s'appelle le **HMaster**, et les nœuds esclaves s'appellent les **RegionsServers**.



Pr. M. Ghazouani

16

9 décembre 2024

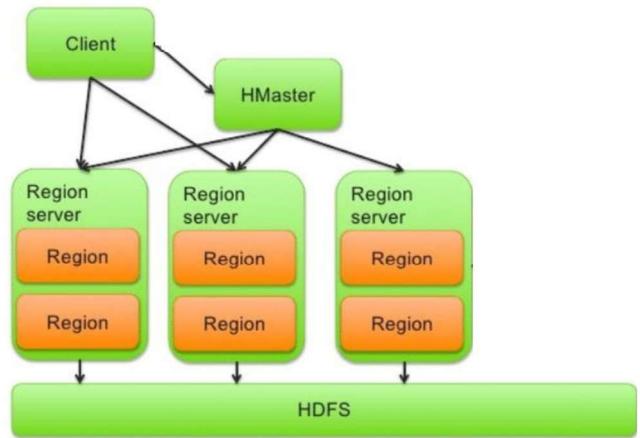
Architecture et fonctionnement du HBase



Le **client** envoie des requêtes pour lire ou écrire des données.

HMaster :

- C'est le chef d'orchestre de HBase.
- Il gère les Region Servers (les serveurs qui contiennent les données).
- Il distribue les régions (une partie des données) sur les serveurs et surveille leur santé.



Pr. M. Ghazouani

17

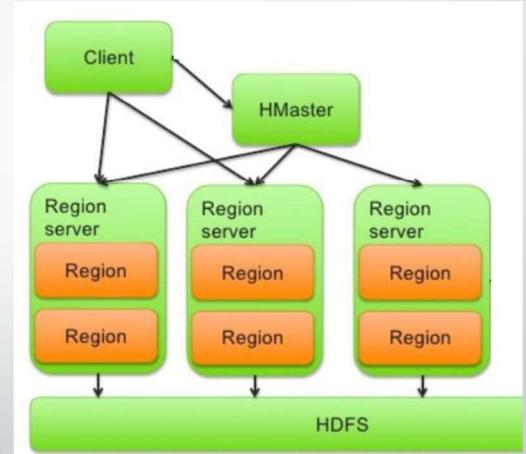
9 décembre 2024

Architecture et fonctionnement du Hbase



Region Servers :

- Ces serveurs stockent les régions, c'est-à-dire des morceaux de données.
- Chaque Region Server peut contenir plusieurs régions.
- La taille de chaque région peut être paramétrée dans un fichier de configuration hbase-site.xml.
- Lorsque la taille d'une région dépasse la taille maximale que vous avez définie dans le fichier de configuration, la région se partitionne automatiquement en deux.



Pr. M. Ghazouani

18

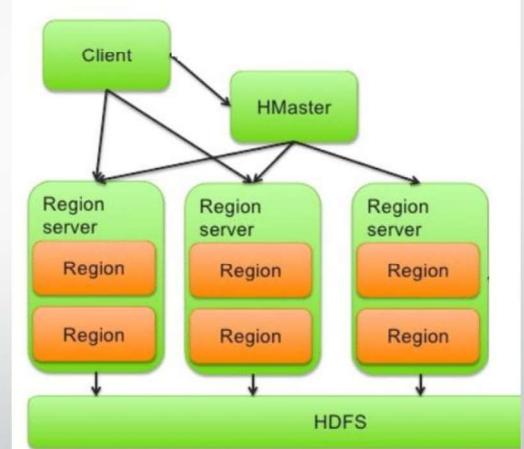
9 décembre 2024

Architecture et fonctionnement du Hbase



Regions :

- Une région est une partie d'une table HBase. Chaque table est divisée en plusieurs régions.
- Les régions sont attribuées à des Region Servers pour équilibrer la charge.

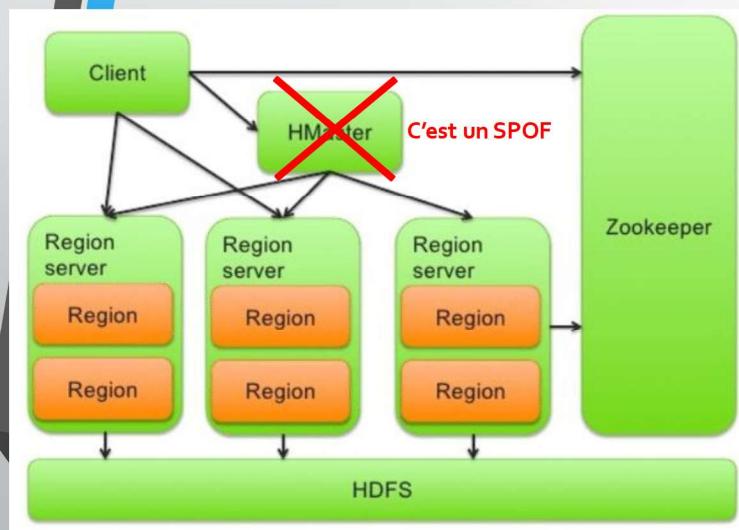


Pr. M. Ghazouani

19

9 décembre 2024

Architecture et fonctionnement du Hbase



Pr. M. Ghazouani

20

- Zookeeper est un projet open-source qui fournit des services tels que la gestion des informations de configuration, la synchronisation distribuée, etc.
- Les clients communiquent avec les serveurs de la région via zookeeper à travers un fichier "**hbase:META**".
- Dans un mode standalone, Hbase prend en charge zookeeper et joue le rôle de zookeeper.

Exploitation d'une table HBase

- Dans les SGBDR, pour interagir avec la Base de données, le SGBDR utilise le SQL et celui-ci sert à la fois de langage de définition de la structure et d'exploitation de la base de données.
- Deux catégories d'opérations y sont fournies : les opérations de définition de la base de données (le CRUD : CREATE, INSERT, UPDATE, DELETE) et les opérations d'interrogation de la base de données (SELECT).
- Il en est de même avec HBase. HBase fournit une série d'opérations pour interagir avec les tables HBase. Ces opérations peuvent être exploitées en masse par le MapReduce et elles sont au nombre de quatre :
 - Get
 - Put
 - Scan
 - Delete

Exploitation d'une table HBase

Get

Get : équivalent à l'opération SQL SELECT, cette opération permet de renvoyer les valeurs soit d'une colonne soit d'une famille de colonnes, soit alors des versions des valeurs de cellule pour une ligne spécifiée.

```
get 'table name', 'rowkey_value', {COLUMN =>
  ['column_family:column_name', ..... , 'column_family:column_name'],
  VERSIONS => <Nombre de versions retournées>...}

get 'Customers', 'CCo9877', {COLUMN => ['personal:customer_name',
  'public: enterprise'], VERSIONS => 10}
```

Cette requête porte sur la table Customers et extrait les valeurs des colonnes customer_name et enterprise pour la Row Key de valeur CCo9877, elle y extrait les 10 premières versions de la row_key

9 décembre 2024



Exploitation d'une table HBase

Put

Put : équivalent à un INSERT en SQL, cette opération ajoute une nouvelle ligne dans la table si cette ligne n'existe pas (c'est-à-dire que la valeur de clé n'existe pas dans la table) ou ajoute une nouvelle version de la même ligne si elle existe déjà dans la table.

Comme avec le GET, le PUT peut se limiter à ajouter une ligne dans une colonne ou une famille de colonnes spécifiques. La syntaxe de cette commande est la suivante :

```
Put 'table_name', 'rowKey_value', 'column_family:column_name',
     'value', ... 'column_family:column_name', 'value'
```

```
Put 'customers', 'CCo9877', 'personal:customer_name', 'juvénal',
     'public:enterprise', 'CAPGEMINI' ;
```

Exemple : ajouter à la table Customers une nouvelle version de la rowkey 'CCo9877', les valeurs respectives 'juvénal' et 'CAPGEMINI' aux colonnes 'personal:customer_name', 'public:enterprise'. Cela donne la requête ci-dessus. 23

Pr. M. Ghazouani

23

9 décembre 2024



Exploitation d'une table HBase

Scan

Scan : cette opération permet d'itérer sur plusieurs lignes pour récupérer les lignes de familles de colonnes, colonnes ou cellules pour des clés spécifiques. La syntaxe de cette commande est la suivante :

```
scan '<TABLE>', {COLUMNS => ['<Colonne1>', ..., '<ColonneN>'],
                   VERSIONS => <nombre de versions retournées>, FILTER => <critère
                   filtre>, LIMIT = <nbre_de_valeurs_renvoyées> ...}
```

Exemple :

```
scan 'customers', {COLUMNS => ['personal:customer_name', 'public:
enterprise'], VERSIONS => 10, FILTER => "PrefixFilter('CCo9877')"}.
```

On peut transférer les résultats de notre requête en la re-écrivant de la façon suivante :

```
echo " scan 'customers', {COLUMNS => ['personal:customer_name',
'public: enterprise'], VERSIONS => 10, FILTER =>
\"PrefixFilter('CCo9877')\" } => results.txt ;"
```

Pr. M. Ghazouani

24

24

Exploitation d'une table HBase

Delete

Delete : à l'instar du DELETE en SQL, cette opération supprime une ou plusieurs lignes spécifiques, colonnes, familles de colonnes ou cellules. La syntaxe pour cette commande est la suivante :

```
DELETE 'table_name>', 'rowKey_value>', 'column_name >', 'time_stamp>'
```

Exemple :

```
DELETE customers 'CCo9877', 'personal: customer_name', 1209201600002345
```

Cette commande supprime la version 1209201600002345 de la row key CCo9877 pour la colonne customer_name ;

Exploitation d'une table HBase

Apache a intégré dans Hive une classe HBaseStorageHandler qui permet de mapper les tables HBase à Hive.

Vous pouvez ainsi utiliser une opération CREATE TABLE dans HiveQL pour créer une table dans HBase, SELECT pour renvoyer les valeurs, INSERT pour insérer les valeurs dans la table HBase et ainsi de suite. supposons que nous voulons créer une table HBase à partir d'un script HiveQL, la requête serait la suivante :

```
CREATE TABLE IF NOT EXISTS Clients(
    id_client STRING,
    concat(Nom_client, ' ', prenom_client) Noms_client
    datediff(date_naissance, CURRENT_DATE()) Age_client,
    Substr(genre, 1, 1) genre,
    Quantité * Prix_unitaire Vente)
    STORED BY 'org.apache.hadoop.hive.HBaseStorageHandler'
    WITH SERDEPROPERTIES ("hbase.columns.mapping" = "id_client:key",
    personal:customer_name, personal:customer_age, personal:customer_gender, product:sales")
    TBLPROPERTIES ("hbase.table.name" = "Customers");
```



27

Travail avec HBase

APACHE HBASE

- Vous pouvez démarrer le shell interactif HBase en utilisant la commande «hbase shell».
- Pour quitter la commande shell interactive à tout moment, tapez exit ou utilisez <ctrl + c>.
- Vérifiez le fonctionnement du shell avant de continuer. Utilisez la commande list à cet effet. List est une commande utilisée pour obtenir la liste de toutes les tables dans HBase.
- **version** : Cette commande renvoie la version de HBase utilisée dans votre système.

9 décembre 2024

Pr. M. Ghazouani

28

28

Travail avec HBase

Creating a Table using HBase Shell

You can create a table using the **create** command, here you must specify the table name and the Column Family name. The **syntax** to create a table in HBase shell is shown below.

```
create '<table name>','<column family>'
```

Example

Given below is a sample schema of a table named emp. It has two column families: "personal data" and "professional data".

Row key	personal data	professional data

You can create this table in HBase shell as shown below.

```
hbase(main):002:0> create 'emp', 'personal data', 'professional data'
```

Travail avec HBase

Disabling a Table using HBase Shell

To delete a table or change its settings, you need to first disable the table using the disable command. You can re-enable it using the enable command. Given below is the syntax to disable a table:

```
disable 'emp'
```

is_disabled

This command is used to find whether a table is disabled. Its syntax is as follows.

```
hbase(main):031:0> is_disabled 'emp'
```

disable_all

This command is used to disable all the tables matching the given regex. The syntax for disable_all command is given below.

```
hbase> disable_all 'r.*'
```

Travail avec HBase

Enabling a Table using HBase Shell

Syntax to enable a table:

```
enable 'emp'
```

is_enabled

This command is used to find whether a table is enabled. Its syntax is as follows:

```
hbase> is_enabled 'table name'
```

The following code verifies whether the table named **emp** is enabled. If it is enabled, it will return true and if not, it will return false.

```
hbase(main):031:0> is_enabled 'emp'  
true  
0 row(s) in 0.0440 seconds
```

Travail avec HBase

describe

This command returns the description of the table. Its syntax is as follows:

```
hbase> describe 'emp'
```

alter

Alter is the command used to make changes to an existing table. Using this command, you can change the maximum number of cells of a column family, set and delete table scope operators, and delete a column family from a table.

Changing the Maximum Number of Cells of a Column Family

Given below is the syntax to change the maximum number of cells of a column family.

```
hbase(main):003:0> alter 'emp', NAME => 'personal data', VERSIONS => 5
```

Travail avec HBase

alter

```
alter 'test',{METHOD=>'table_att', READONLY=>true}
```

Table Scope Operators

Using alter, you can set and remove table scope operators such as MAX_FILESIZEx, READONLY, MEMSTORE_FLUSHSIZE, etc.

Setting Read Only

Below given is the syntax to make a table read only.

```
hbase(main):006:0> alter 'emp',METHOD=>'table_att', READONLY=>true
```

Removing Table Scope Operators

```
hbase> alter 'emp',METHOD=>'table_att_unset', NAME=>READONLY
```

Deleting a Column Family

Using alter, you can also delete a column family. Given below is the syntax to delete a column family using alter.

```
hbase> alter 'table name ', 'delete' => 'column family '
```

Travail avec HBase

Dropping a Table using HBase Shell

Using the **drop** command, you can delete a table. Before dropping a table, you have to disable it.

```
hbase(main):018:0> disable 'emp'
```

```
hbase(main):019:0> drop 'emp'
```

Verify whether the table is deleted using the exists command.

```
hbase(main):020:0> exists 'emp'
```

drop_all

This command is used to drop the tables matching the “regex” given in the command. Its syntax is as follows:

```
hbase> drop_all 't.*'
```

Travail avec HBase

exit

You exit the shell by typing the **exit** command.

```
hbase(main):021:0> exit
```

Stopping HBase

To stop HBase, browse to the HBase home folder and type the following command.

```
stop-hbase.sh
```

Travail avec HBase

Inserting Data using HBase Shell

PUT

Using **put** command, you can insert rows into a table. Its syntax is as follows:

```
hbase(main):005:0> put 'emp','1','personal data:name', ' Said'
```

```
hbase(main):005:0> put 'emp','1','personal data:city', ' Casa '
```

```
hbase(main):005:0> put 'emp','1','professional data:function', ' Manager'
```

```
hbase(main):005:0> put 'emp','1','professional data:salary', '10000 '
```

Roy key	personnal data		professional data	
empid	name	city	function	salary
1	Said	Casa	Manager	10000
2	Fahd	rabat	Technicien	6000
3	Karim	Tanger	enginner	9000

Travail avec HBase

Updating Data using HBase Shell

PUT

You can update an existing cell value using the **put** command. To do so, just follow the same syntax and mention your new value as shown below.

```
hbase(main):005:0> put 'emp','1','personal data:city', 'Casablanca '
```

Travail avec HBase

Reading Data using HBase Shell

The **get** command and the **get()** method of **HTable** class are used to read data from a table in HBase. Using **get** command, you can get a single row of data at a time. Its syntax is as follows:

```
hbase(main):012:0> get 'emp', '1'
```

Reading a Specific Column

Given below is the syntax to read a specific column using the **get** method.

```
hbase(main):015:0> get 'emp', '1', {COLUMN => 'personal data:name'}
```

Travail avec HBase

Scanning using HBase Shell

The **scan** command is used to view the data in HTable. Using the scan command, you can get the table data. Its syntax is as follows:

```
hbase(main):010:0> scan 'emp'
hbase(main):010:0> scan 'emp', {RAW => true, VERSIONS => 3}
```

Count

You can count the number of rows of a table using the **count** command. Its syntax is as follows:

```
hbase(main):023:0> count 'emp'
```

Truncate

This command disables, drops and recreates a table. The syntax of **truncate** is as follows:

```
hbase(main):011:0> truncate 'emp'
```

Travail avec HBase

Deleting a Specific Cell in a Table

Using the **delete** command, you can delete a specific cell in a table. The syntax of **delete** command is as follows:

```
hbase(main):006:0> delete 'emp', '1', 'personal data:city'
```

Deleting All Cells in a Table

Using the “**deleteall**” command, you can delete all the cells in a row. Given below is the syntax of **deleteall** command.

```
hbase(main):007:0> deleteall 'emp','1'
```



41



42