

TP4 : HBase*

Pierre Monnin, Guillaume Mérroué, Célian Ringwald, Nicolas Robert
{pierre.monnin, guillaume.meroue, celian.ringwald, nicolas.robert}@inria.fr

2024 – 2025

Objectif

HBase est une base de données orientée colonnes: les données sont stockées sous forme de colonnes (Column) et non pas de lignes (Row) comme pour Oracle ou MySQL par exemple. Un cluster HBase peut stocker plusieurs milliards de lignes ainsi que plusieurs millions de colonnes pour chaque ligne. Les colonnes sont regroupées par famille de colonne (Column Family), Hbase supportant plusieurs familles de colonne par table, même s'il n'est pas conseillé d'en créer plus de 3 par table. Chaque ligne est composée d'une clé (Row Key) et de plusieurs colonnes, qui peuvent être totalement différentes entre deux lignes d'une même table.

Dans ce TP sur HBase, vous apprendrez à utiliser les commandes de base, et vous travaillerez sur la modélisation d'un problème à résoudre en utilisant HBase. Vous pourrez également utiliser l'API Java ou Python pour interagir avec HBase.

Software

Pour utiliser HBase, nous allons utiliser l'image Docker suivante :

```
1 docker run -ti harisekhon/hbase
```

Attention, cette commande lance HBase et ouvre directement une CLI HBase à l'intérieur. N'oubliez pas de faire un mapping de port si vous souhaitez interagir avec HBase via Python ou Java.

Documentation

- HBase : <https://hbase.apache.org/book.html>
- HBase CLI : https://www.tutorialspoint.com/hbase/hbase_shell.htm
- Docker Hbase : <https://hub.docker.com/r/harisekhon/hbase>

Utilisation du shell HBase

1. Créez une table nommée <votre-nom>-shell avec un seule famille de colonne, nommée cf. Pour cette opération, vous pouvez utiliser la commande `create`. Vérifiez que votre table a bien été créée via la commande `list`.
2. Insérez les lignes suivantes dans votre table, via la commande `put`

*D'après le TP proposé par Sabeur Aridhi

```
1 jean-dupont, {nom: Dupont, prenom: Jean, ville: Nancy }
2 marie-moine, {nom: Moine, prenom: Marie, ville: Paris }
3 marc-pinault, {nom: Pinault, prenom: Marc, ville: Nancy }
4 romain-guillet, {nom: Guillet, prenom: Romain, ville: Lyon }
5 jeanne-lenglet, {nom: Lenglet, prenom: Jeanne, ville: Lyon }
6 lisa-magnant, {nom: Magnant, prenom: Lisa, ville: Marseille }
```

3. Listez les lignes de votre table précédemment créée via la commande `scan`
4. Récupérez les informations sur l'utilisateur `jean-dupont` dans votre table via la commande `get`
5. Récupérez la ville des utilisateurs dont le prénom commence par `ma` dans votre table
6. Suite à la mise en place du RGPD, vous avez reçu deux demandes :
 - Une demande de Lisa Magnant demandant la suppression de l'ensemble de ses informations
 - Une demande de Jeanne Lenglet demandant la suppression des informations concernant sa ville

Trouvez les bonnes commandes pour procéder à ces requêtes.

Modélisation

Un client vient vous rencontrer afin de développer un serveur de chat. Le serveur doit disposer de plusieurs channels, qui peuvent être créés dynamiquement par les utilisateurs. En outre, un utilisateur peut rejoindre un channel via son nom. Pour un utilisateur, son pseudonyme et son adresse e-mail doivent être stockés. Les channels rejoints par un utilisateur (et la date à laquelle il les a rejoints) doivent également être stockés. Pour un channel, l'ensemble de ses messages doivent être stockés. Un message contient les champs suivants : sa date, précise à la milliseconde, l'utilisateur l'ayant créé et le contenu du message. Le client n'a qu'une seule exigence : il veut que vous utilisiez HBase comme base de données pour ce projet.

1. Combien de tables sont nécessaires à la réalisation de ce projet ?
2. Pour chaque table, listez les familles de colonnes nécessaires.
3. Pour chaque table, détailler la clé utilisée pour les lignes. Pour rappel, lors d'un `scan`, HBase ordonne les résultats par clé, puis par nom de colonne.
4. Comment lister tous les channels disponibles ?
5. Comment lister les channels dont un utilisateur est membre ?
6. Comment lister les informations d'un utilisateur ?
7. Comment lister les 20 derniers messages d'un channel, ordonnés par timestamp inverse ?
8. Comment permettre à un utilisateur de modifier son message ?
9. Comment lister l'historique de modification d'un message ?

Utilisation de l'API Java (ou Python)

1. Exécutez l'ensemble des opérations de la première section en Java (ou Python)
2. Implémentez les fonctions de la partie modélisation en Java (ou Python)