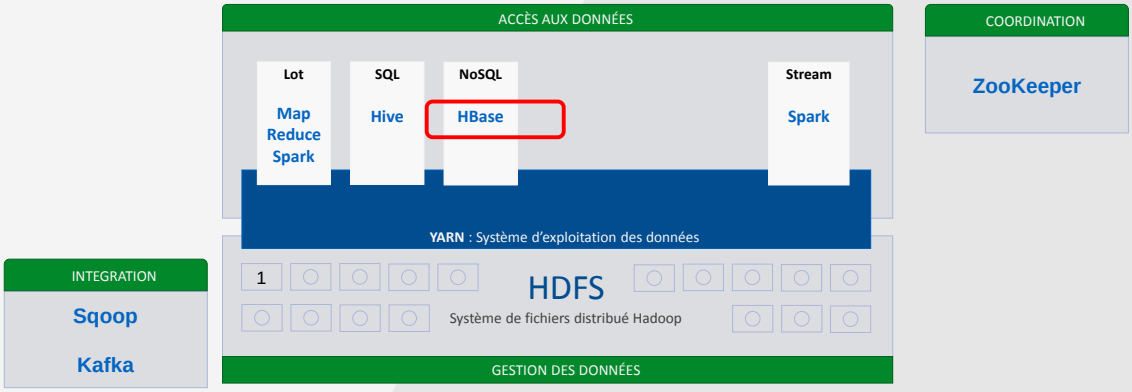
L'écosystème Hadoop



Qu'est ce que c'est Hbase

- Hbase est une base de données
  - Distribuée
  - Orientée colonnes
  - Construite au dessus HDFS
  - Open source
  - Scalable horizontalement

- Hbase fait partie de l'écosystème Hadoop qui fournit un accès d'écriture/lecture temps réel sur HDFS

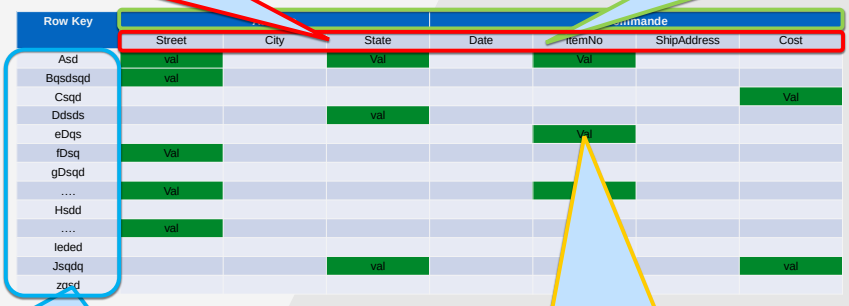
Modèle de données HBase

Composant	Description
Table	Les données sont organisées en tables, chacune contient une collection de lignes (Row)
Row key	Les lignes d'une table sont identifiées par une clé
Column Family	Les colonnes sont regroupées en column families
Column Qualifier	Chaque colonne est identifiée par un qualifier
Cell	Une combinaison de row key, column family, column qualifier et version. Elle contient une valeur
Version	Les valeurs d'une cellule sont versionnées

Modèle de données Hbase: composants

- Chaque colonne est identifiée par un qualifier
- Les lignes peuvent ne pas contenir les mêmes colonnes

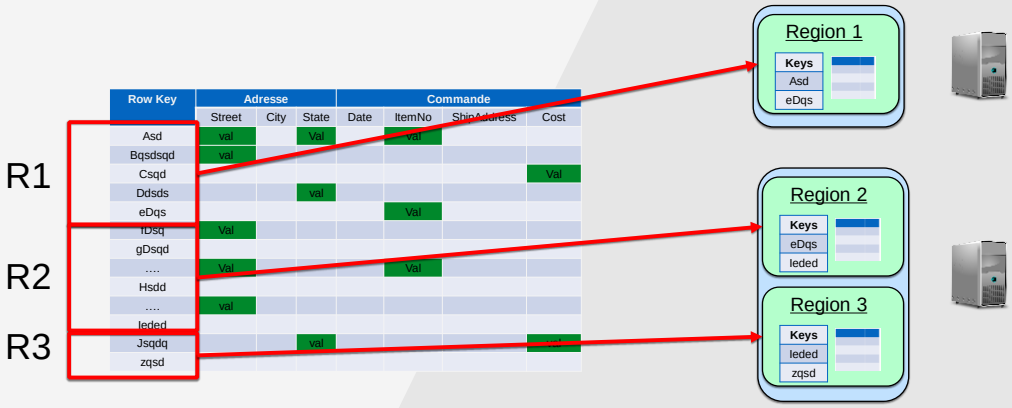
- Les colonnes sont regroupées en column families
- Les column_families font partie de la description du schéma



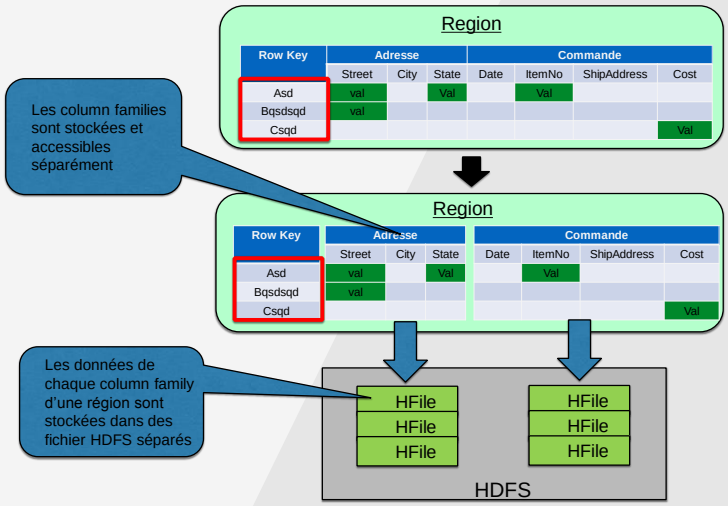
- Row key identifie les lignes de la table
- Les clés sont ordonnées lexicographiquement à la base de leurs représentation en binaire

- Une combinaison de row key, column family, column qualifier et version. Elle contient une valeur.
- Chaque cellule est versionnée

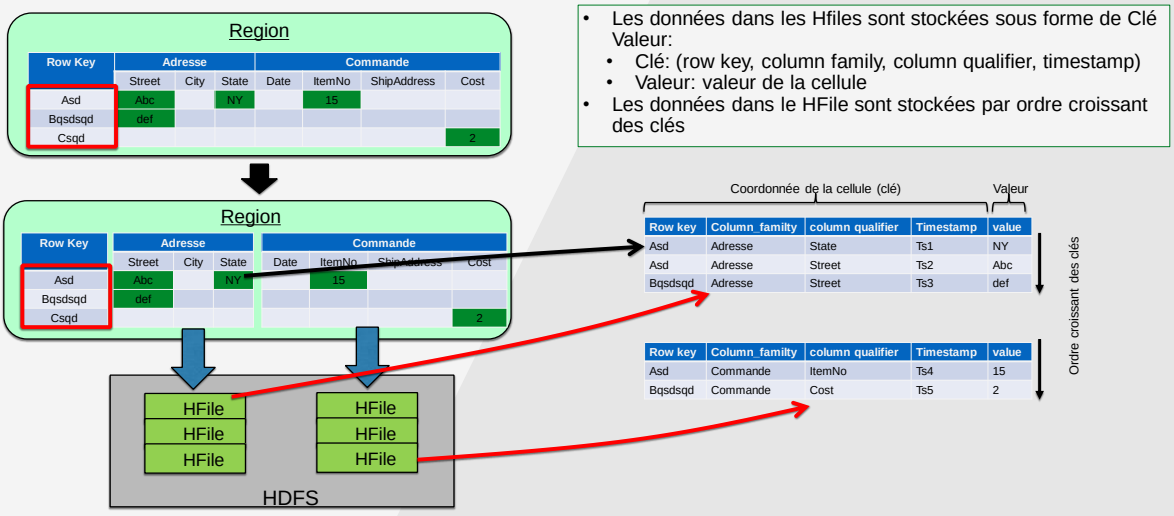
Modèle de données Hbase: Régions



Modèle physique de données (1)



Modèle physique de données (2)

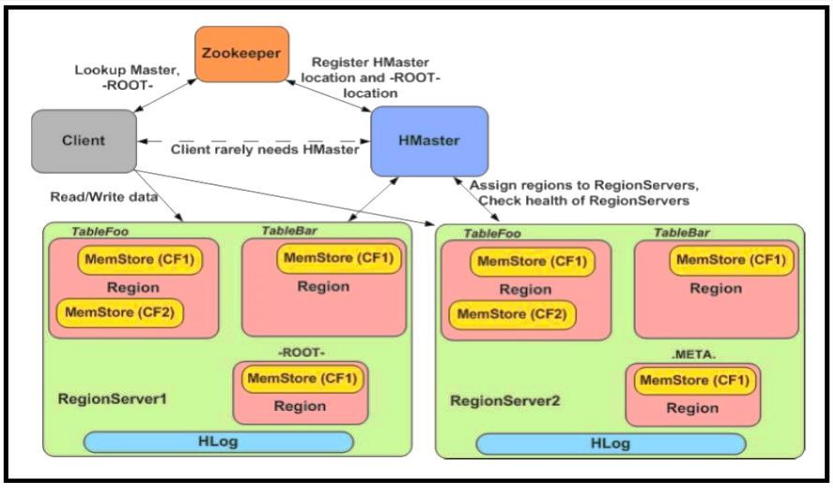


- Les données dans les Hfiles sont stockées sous forme de Clé Valeur:
  - Clé: (row key, column family, column qualifier, timestamp)
  - Valeur: valeur de la cellule
- Les données dans le HFile sont stockées par ordre croissant des clés

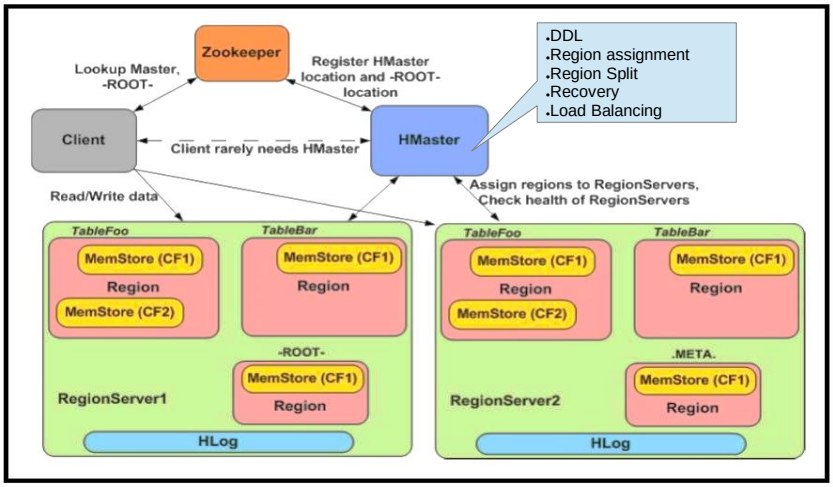
HBase: opérations basiques

Opération	Description
GET	Récupérer une ligne
PUT	Insérer une donnée dans une ligne (ajout ou mise à jour)
SCAN	Récupérer une plage de clés
DELETE	Supprimer une ligne ou une plage de clés

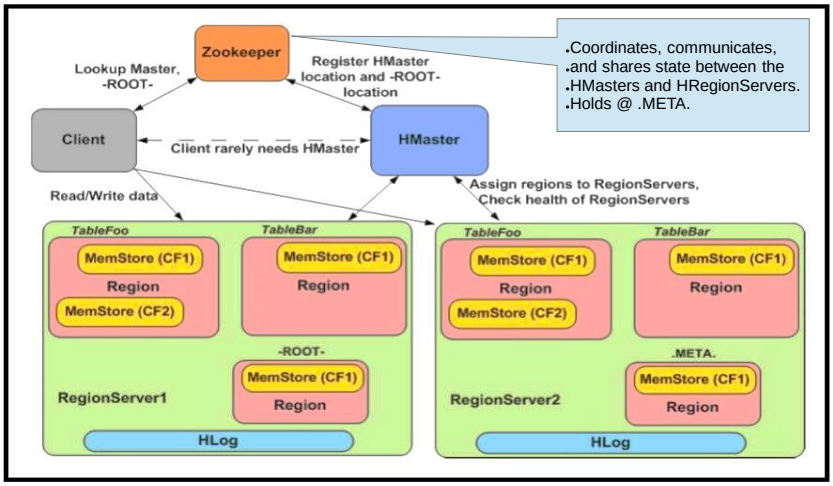
HBase Architecture



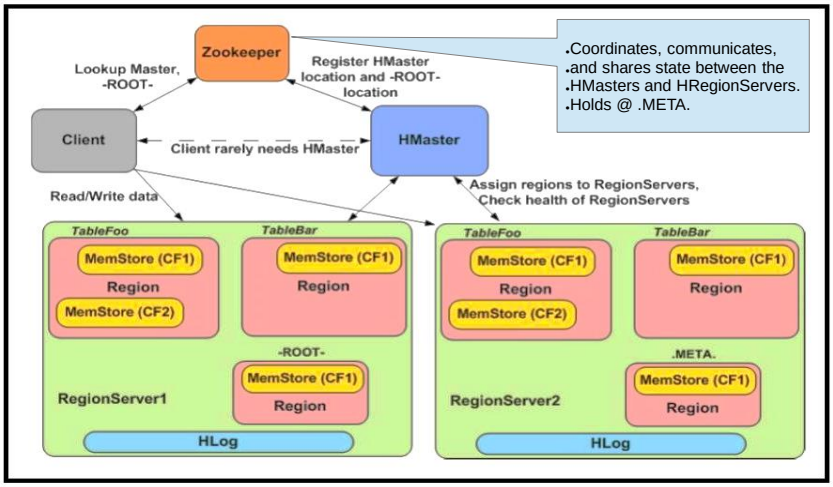
HBase Architecture : HMaster



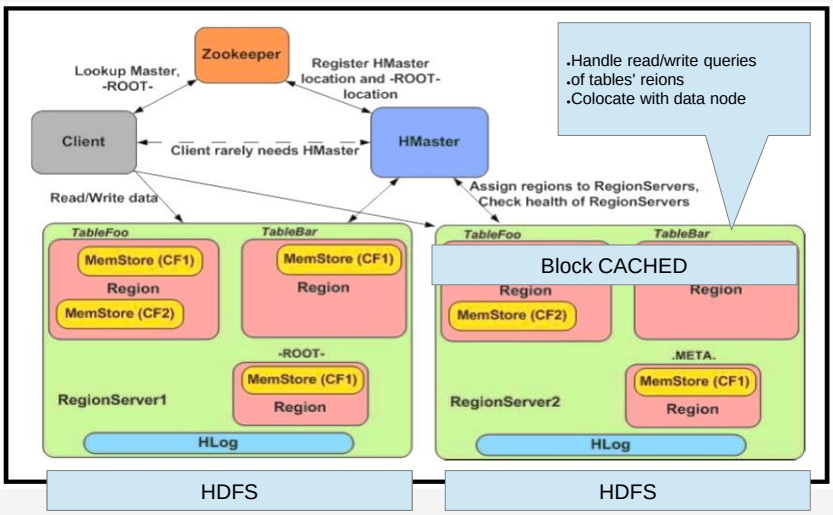
HBase Architecture : ZK



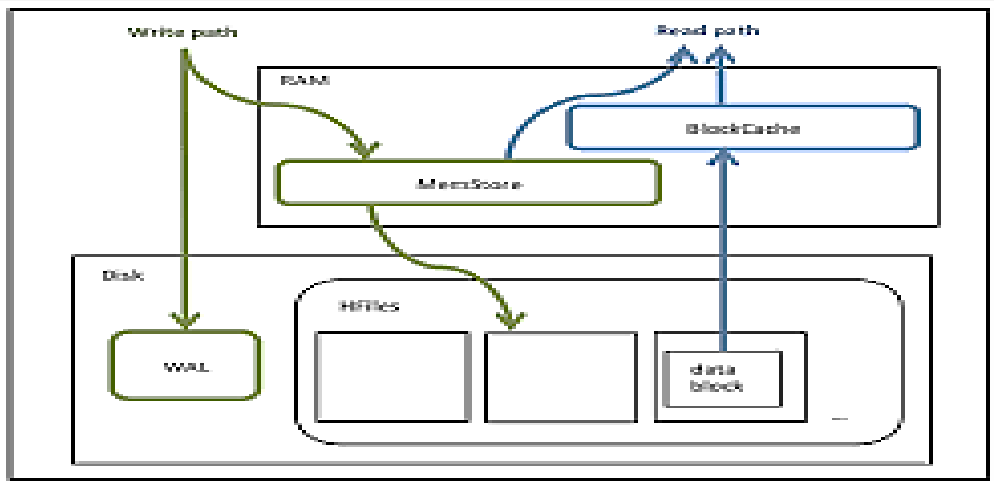
HBase Architecture



HBase Architecture : Region Server



Processus de lecture et écritureans HBase

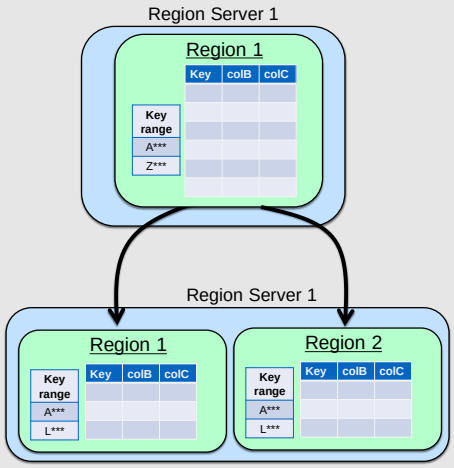


HBase & SGBDR

	SGBDR	HBase
Modèle de données	Fixé	Flexible
Langage de requêtes	SQL	Get/put/scan/delete
Transactions	Multi-tables/multi-lignes	Une ligne uniquement
Jointure	Oui	Non
Taille de données max	Térabytes	Milliers de pétabytes
Vitesse de lecture/écriture	1000 req/seconde	Millions de req/seconde
Scalabilité	Après coup	Conçue pour
Tolérance aux pannes	Maitre-esclave	Conçue pour

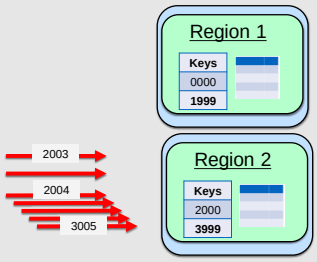
Split des régions HBase

- Les régions HBase se divise (split) quand elles atteignent une certaine taille
  - La taille de split est configurable
- Le région server divise une région dés que sa taille atteint un certain seuil
- Chaque région est divisée en 2 régions filles
- Chaque région fille s'occupe de la moitié de la plage des clés de la région mère
  - La clé de division est la clé qui divise l'espace des clés possibles de la région mère en deux
- Les régions filles sont ouvertes (pour écriture et lecture) simultanément sur le même server
- L'opération de split est reportée au master
- Une opération de load-balancing peut être déclenché par le Master



Split des régions et Hot-Spotting

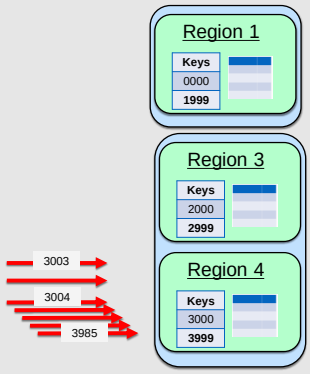
- Hot-Spotting est le fait que la majorité des écritures vont vers la même région
- Le Hot-Spotting arrive quand la distribution des clés n'est pas uniforme sur toutes les région
  - Exemple: quand la clé est un timestamp



La région 2 va splitter vu qu'elle grandit

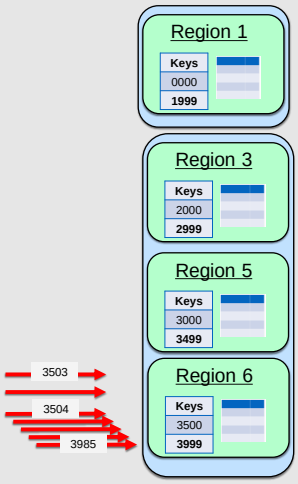
Split des régions et Hot-Spotting

- Hot-Spotting est le fait que la majorité des écritures vont vers la même région
- Le Hot-Spotting arrive quand la distribution des clés n'est pas uniforme sur toutes les région
  - Exemple: quand la clé est un timestamp
- Le hot-spotting provoque des Splits de régions trop intempestifs



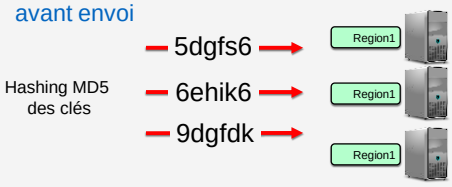
Split des régions et Hot-Spotting

- Conséquences:
  - Dégradation des performances vu que Hbase passe plus de temps à splitter les régions
  - Les ressources du cluster sont mal exploitées vu que certains nœuds peuvent être saturés tant dis que d'autres ne font rien

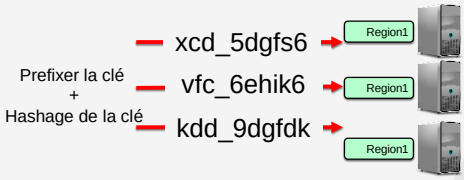


Conception de schémas: hotspotting

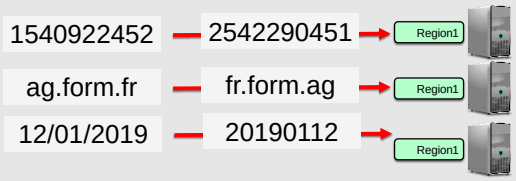
1. Random key: hashage de la clé avant envoi



2. Salting de la clé: préfixer la clé + hashage avant envoi

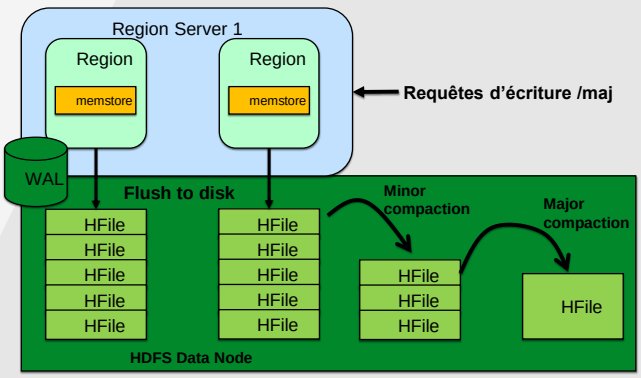


3. Inverser les clés de types numériques, dates, timestamps, nom de domaine ou ayant une tailles fixe



Compactions dans HBase

- Minor compaction
  - Combine un nombre configurable de HFiles en un seul HFile plus large
  - Le nombre de Hfile à compacter et la fréquence de compaction sont configurables
  - Les compactions mineurs sont importantes pour réduire le nombre de lecture disques à faire pour la lecture d'une cellule
- Major compaction
  - Combine tous les HFiles d'une region en un seul et large HFile
  - Supprime toutes les cellules marquées pour la suppression
  - Elle est consommatrice en I/O
  - La compaction majeure doit être programmée en dehors des heures de charge de HBase

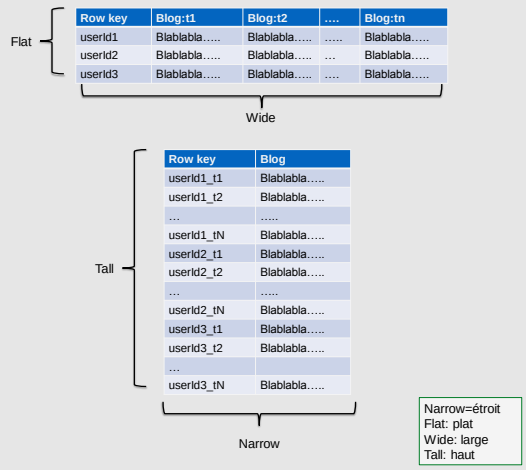


Conception de schémas: guidelines

- Avant la conception d'un schema, toutes les requêtes d'accès doivent être identifies
  - Partir des requêtes d'accès pour concevoir les schémas
  - Un schema par application
  - Un schema HBase ne peut pas être performant pour tous les cas d'usage
  - Dupliquer les données si necessaire pour booster le performances
- Concevoir un schema de telle sorte que les données qui sont accessibles ensemble soient stockées dans le même endroit
- Dénormalisation des données:
  - Pas de jointure dans Hbase
- Limiter le nombre de column families
  - HBase a du mal à gérer des tables ayant plus de 3 column families
- Conception des rowkeys:
  - Eviter les clés monotiquement croissante
  - Minimiser la taille des clés et des colonnes sans pour autant concevoir des clés inutiles pour le get et le scan
- Garder le nom des colonnes aussi concis que possible
- Les rowkeys ne sont pas updatables:
  - Si une clé est à changer, il faut réécrire toutes les données qui lui sont rattachées
- Eviter l'usage de TTL

Conception de schémas: Table Tall-Narrow vs Flat-Wide

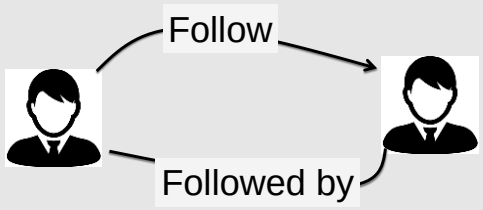
- Tall-Narrow:
  - Une table avec peu de colonnes et beaucoup de lignes
  - Stocke une partie de la cellule dans la clé
  - Une requête par rowkey va sauter certains Hfile
  - Ça scale mieux que Flat (vu le sharding se fera aux frontières des blogs)
  - Pas de transactions possibles: il n'y a pas de transaction
- Flat-wide:
  - Une table avec beaucoup de colonnes et peu de lignes
  - Elle est atomique au niveau de la row
  - Les requêtes doivent contenir le nom de colonnes ce qui ne sautera pas certains HFiles et row keys



Conception de schéma: étude de cas (1)

Gestion des relations entre utilisateurs de Twitter

1. Description: stockage des relations de follow et followed-by existantes sur Twitter
2. Requêtes auxquelles il faut répondre
   - Quels sont les followers d'un user ?
   - Quels sont les personnes qu'un user follow ?
   - Est-ce qu'un user A follow un user B ?



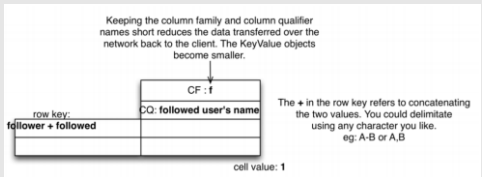
Conception de schéma: étude de cas (1)

Gestion des relations entre utilisateurs de Twitter

1. Description: stockage des relations de follow et followed-by existantes sur Twitter
2. Requêtes auxquelles il faut répondre
   - Quels sont les followers d'un user ?
   - Quels sont les personnes qu'un user follow ?
   - Est-ce qu'un user A follow un user B ?

Follow
Followed by

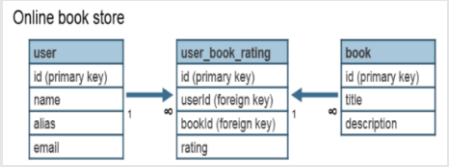
	follows			
AK	foo:1	bar:1	baz:1	troy:1
foo	bar:1	AK:1		



Conception de schéma: étude de cas (2)

Système de notation de livres dans une bibliothèque

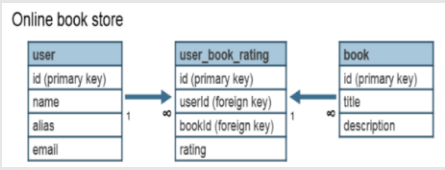
1. Description: stocker les notations de livres empruntés par des usages de bibliothèques
2. Requêtes auxquelles il faut répondre:
   - Trouver les livres et les notations données par un utilisateur X
   - Trouver les utilisateurs ainsi que les notations données pour un livre Y



Conception de schéma: étude de cas (2)

Système de notation de livres dans une bibliothèque

1. Description: stocker les notations de livres empruntés par des usages de bibliothèques
2. Requêtes auxquelles il faut répondre:
   - Trouver les livres et les notations données par un utilisateur X
   - Trouver les utilisateurs ainsi que les notations données pour un livre Y



User table Column family for book ratings by userid for bookids

Key	data:fname	...	rating:bookid1	rating:bookid2
userid1			5	4

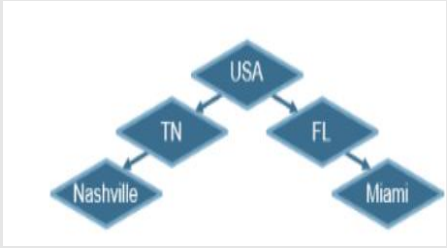
Book table Column family for ratings for bookid by userid

Key	data:title	...	rating:userid1	rating:userid2
bookid1			5	4

Conception de schéma: étude de cas (3)

Stockage de données hiérarchiques

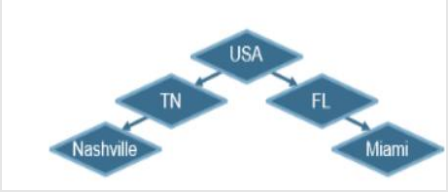
1. Description: stocker des données hiérarchiques (représentées sous forme d'un arbre)
2. Requêtes auxquelles il faut répondre:
   - Trouver les enfants d'un noeud
   - Trouver les parents d'un noeud



Conception de schéma: étude de cas (3)

Stockage de données hiérarchiques

1. Description: stocker des données hiérarchiques (représentées sous forme d'un arbre)
2. Requêtes auxquelles il faut répondre:
   - Trouver les enfants d'un noeud
   - Trouver les parents d'un noeud et retrouver la nature de son père (pays, état, ville, …)
   - Trouver qu'el



Key	Pere:USA	Pere:TN	Pere:FL	Fils:TN	Fils:FL	Fils:Nashville	Fils:Miami
USA				etat	Etat		
TN	pays					ville	
FL	Pays						ville
Nashville		Etat					
Miami			Etat				

Conception de schéma: étude de cas (4)

Stockage de données de type timeseries

1. Description: stocker les données provenant d'appareils médicaux (rythme cardiaque, température, tension, ….)
2. Requêtes auxquelles il faut répondre:
   - Trouver pour un patient donné, tous les prélévements effectués dans l'ordre de leurs prélévement
   - Trouver tous les prélévements entre deux timestamps

Conception de schéma: étude de cas (4)

Stockage de données de type timeseries

1. Description: stocker les données provenant d'appareils médicaux (rythme cardiaque, température, tension, ….)
2. Requêtes auxquelles il faut répondre:
   - Trouver pour un patient donné, tous les prélévements effectués dans l'ordre de leurs prélévement
   - Trouver tous les prélévements entre deux timestamps

Key	Event:temperature	Event:rythme_cardiaque	Event:tension
Patient1_ts1	10	89	13
Patient2_ts1	11	76	12