

## FORMATION APACHE CASSANDRA

---

DUREE : 2 JOURS (14 HEURES)

Cette formation de 2 jours vous apportera tous les éléments dont vous avez besoin pour adopter avec succès Apache Cassandra. Elle est ponctuée d'échanges théoriques et d'ateliers Hands on. Vous aborderez notamment les sujets suivants: Apache Cassandra, Apache Hadoop, Apache Solr, DataStax Enterprise et DataStax OpsCenter.

### PROGRAMME

---

#### LES BASES

- Comprendre ce que sont les colonnes Cassandra (ce qu'il y a après « C\* »), leurs composants et leurs rôles ;
- Décrire le rôle d'une colonne dans une famille de colonne ;
- Identifier les colonnes, composants (TimeStamp, Name/Value, TTL, composants requis ou optionnels) ;
- Détails pour le nom de la colonne ;
- Détails pour la valeur de la colonne ;
- A quelle fin le TimeStamp est-il utilisé ?
- Qu'est-ce que le TimeStamp ?
- Les détails pour le TTL ;
- Que sont les SuperColumns ?
- Que sont les CounterColumns ?
- Comment les colonnes sont triées ?
- Comment les valeurs des colonnes sont validées ?
- A quelle fin les comparateurs sont utilisés ?
- Décrire ce qu'est un comparateur ?
- Que signifie inverser un comparateur ?

#### FAMILLE DE COLONNES

- Qu'est-ce qu'une famille de colonne ?
- Qu'est-ce qu'un « Keyspace » ?
- Qu'est-ce qui est stocké dans un « Keyspace » ?
- Qu'est-ce qu'un cluster ?
- Acquérir une compréhension du « Consistent Hashing »
- Comment les partitions sont sélectionnées
- Que sont les lignes (« rows ») ?
- Qu'est-ce que la « dénormalisation » (« denormalization ») ?
- Qu'est-ce que l'« Eventual Consistency »
- Décrire des familles de colonne statiques : Static Column Families
- Décrire des familles de colonne dynamiques : Dynamic Column Families
- Utiliser de schémas de requêtes efficaces avec C\*
- Identification des anti-modèles dans les requêtes C\*
- Etre capable de « dénormaliser » vos données en C\*
- Que font les TTL ?
- Comment les comparateurs influencent la modélisation des données ?
- Reconnaître un « Cien side join » et comment l'éviter ;
- Savoir comment C\* représente les relations dans les données (il ne le fait pas) ;
- Faire des analyses coût/ bénéfice pour choisir « reads » ou « writes » ;
- Comprendre ce qui se produit quand on insert des données en l'absence de contraintes primaires clés ;
- Comprendre ce que les UUIDs sont et leurs différents types ;

## FORMATION APACHE CASSANDRA

---

DUREE : 2 JOURS (14 HEURES)

### L'API

- Comprendre l'objectif de l'API Thrift ;
- Qu'est-ce que Thrift ?
- Pour quoi utilise-t-on Thrift ?
- Identifier ce qui peut mal se passer et pourquoi ?
- Décrire les besoins de cohérence pour « reads » et « writes » ;
- Décrire les fonctions des méthodes Thrift pour « get », « set », « multiget », « indexed slice »
- A quelle fin la commande « execute\_cql\_query » est utilisée ?
- Comprendre ce qu'il y a dans le type Thrift pour ColumnOrSupercolumn ;
- Qu'est-ce qu'un « slice predicate » ;
- Quelles sont les contraintes disponibles ;
- Qu'est-ce qu'un « key range »
- Pourquoi vous ne devriez pas utiliser « raw Thrift »
- Comment Thrift accède à CQL ?
- Quel client est le mieux pour vous et comment en choisir un ?

### INDEXER

- Expliquer comment C\* utilise "Primary Indexes" ou "Partition Keys" pour localiser les données dans le cluster ;
- Expliquer comment C\* utilise "Primary Indexes" pour localiser les données dans la machine ;
- Apprenez pourquoi vous devriez choisir le « Random Partitioner » ;
- Que sont les indices secondaires ?
- Quelle est la limite absolue de la colonne ?
- Qu'est-ce qu'une limite de colonne plus pratique ?
- Pour quoi les schémas sont-ils bons ?
- Pourquoi les personnes utilisent typiquement les « wide row » dans cassandra ?
- Apprendre comment les lignes sont stockées dans les partitions ;
- Comprendre comment créer et maintenir des indices « wide row » pour : Grouping Sorting Range Queries ;
- Comprendre comment les Super et Composite columns trient les données ;

### EXAMINER UNE APPLICATION CASSANDRA

- Analyser une application Cassandra existante ;
- Identifier des applications efficaces de « Column Families » ;
- Identifier les familles de colonne statiques et dynamiques et comment elles sont utilisées ;
- Identifier les contre-modèles d'application C\* ;
- Utiliser CQL3 pour « dénormaliser » un schéma et éliminer les raccords côté client ;

### COLONNES COMPOSITES

- Que sont les colonnes composites dans Thrift ?
- Dans CQL comment faut-il utiliser les composites ?
- Qu'est-ce qui doit être évité ?

### LISTES ORDONNEES RANGS ESPACES

- Démultiplier les colonnes pour créer des listes ordonnées ;
- Comprendre comment les colonnes sont ordonnées ;
- Comprendre ce qui se produit lorsqu'on change une valeur dans une colonne ;

## FORMATION APACHE CASSANDRA

---

DUREE : 2 JOURS (14 HEURES)

- Comprendre ce qui se produit lorsqu'on insert un nouveau nom de colonne (ou un objet avec un nouveau nom de colonne) ;
- Décrire ce qui se produit lorsqu'on essaye de « mettre à jour » le nom d'une colonne dans une liste (insérer un nouvel objet) ;
- Connaître des manières alternatives de mettre à jour des listes (créer de nouvelles listes, versions, clés de substitution) ;
- Comment utiliser des index inversés pour ajouter, actualiser, effacer des objets d'une liste ;

### INDICES SECONDAIRES

- Comprendre comment des index secondaires sont implémenter dans C\* ;
- Identifier des modèles de requête efficaces avec des index secondaires ;
- Identifier des anti-modèles avec des index secondaires ;
- Etre capable de créer un index secondaire dans un tableau ;
- Etre capable de faire des requêtes dans un index secondaire ;
- Comprendre comment la cardinalité affecte les index secondaires de requête ;

### DONNEES TEMPORELLES (TIME SERIES)

### REUSSITES DE MODELISATION

### HADOOP

### HIVE

### PIG

### SOLR.