Autor: Miguel Alexander Chitiva Díaz

Taller HBase

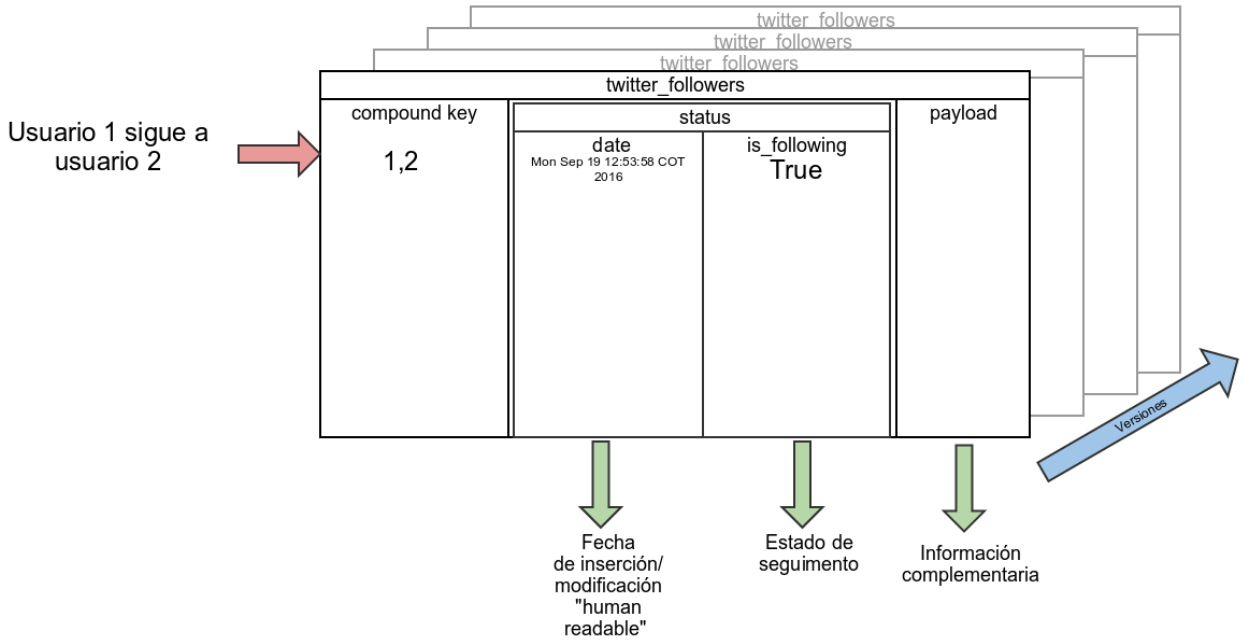
La red social Twitter puede ser entendida como una comunidad de usuarios que siguen y son seguidos por otros usuarios. La relación de seguimiento no es simétrica, es decir, si un usuario
A sigue a un usuario B, no implica que el usuario B siga al usuario A. Como parte de su sistema, Twitter Inc. permite al usuario seguir o dejar de seguir usuarios, consultar la lista de usuarios que sigue, consultar la lista de usuarios que lo siguen.

Últimamente Twitter está contratando personal experto para mejorar su sistema de base de datos y como parte del proceso de entrevista te ha encargado la tarea de diseñar un modelo de datos en HBase que permita de manera eficiente realizar los siguientes patrones de acceso:

Lecturas:
- Listar usuarios que sigue el usuario X.
- Determinar si un usuario X sigue al usuario Y.
- Listar los usuarios que siguen al usuario X.

Escrituras:
- Usuario X sigue un nuevo usuario Y.
- Usuario X deja de seguir un usuario Y.

Modelo de datos propuesto:



Cada fila consta de una *llave compuesta* (Usuario1 sigue a Usuario2) representada por la tupla (1,2), la cual tiene asociados los valores de dos familias de columnas *status, payload* donde la

primera familia almacena el estado de la relación entre los usuarios y la segunda familia representa información complementaria que se requiera guardar que caracterice la relación en el tiempo.

status:date almacena la fecha de inserción o modificación del estado de la relación entre un par de usuarios, en un formato simple de leer.
status:is_following almacena el estado de la relación entre dos usuarios, *true* significa que existe la relación entre y en un periodo de tiempo , es decir en el momento en que se crea la relación el valor deberá ser *true* en el momento en que un usuario deje de seguir a otro no existirá la operación de borrado de la base de datos, en lugar de ello el estado de la relación cambia a *false* indicando que la relación no existe actualmente.

De tal forma que se aprovecha la capacidad de HBASE de versionamiento utilizando los timestamps, esto con el objetivo de que la base de datos tenga la capacidad de realizar análisis a nivel temporal de forma sencilla.

Scripts:
Con la intensión de ejecutar de forma sencilla las soluciones se adjunta una carpeta denominada 'scripts' que contienen todos los scripts que realizan las actividades enunciadas, estos pueden ser ejecutados directamente en el shell de Linux de la siguiente forma:

```
cat .scripts/fileName | pathHbaseBinaries/hbase shell > ./outputfiles/output_name.txt
```

Como se puede apreciar generara un archivo con el nombre 'output_name.txt' con los resultados de la ejecución, originalmente en estos archivos contendrán las salidas de la consola ejecutados desde mi equipo.

Ejemplo:

```
..ropbox/Self_Learning/BigData❱ cat ./scripts/hbase_populate | /opt/hbase-1.2.3/bin/hbase shell > ./ouputfiles/output_populate.txt
2016-09-19 15:16:07,160 WARN  [main] util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
file:/opt/hbase-1.2.3/lib/jruby-complete-1.6.8.jar!/builtin/javasupport/core_ext/object.rb:99 warning: already initialized constant Date
..ropbox/Self_Learning/BigData❱
```

1. **Crear esquema:**

   ```
   create 'twitter_followers', {NAME => 'status', VERSIONS => 20}, {NAME => 'payload', VERSIONS => 20}
   ```

   Crea la tabla denominada 'twiter_followers' las cuales posee las familias de columnas 'status' y 'payload' las que manejaran hasta 20 versiones.

2. **Usuario X sigue a usuario Y:**

```
import java.util.Date
put 'twitter_followers', '1,2', 'status:date', Date.new().toString()
put 'twitter_followers', '1,2', 'status:is_following', 'True'
```

En la línea 1 se importa la librera Date de java con el fin de utilizarla para almacenar las fechas de creación/modificación en un formato de fácil lectura. En la línea 2 se introduce la fecha de creación de la relación entre usuario 1 y 2. En la línea 3 se introduce el estado de la relación, 'True' para indicar que la relación está activa.

3. **Usuario X deja de seguir un usuario Y:**

```
put 'twitter_followers', '1,4', 'status:is_following', 'False'
```

Cuando un usuario deja de seguir a otro simplemente se pone el estado de seguimiento en 'False' aunque si no se desea conservar la capacidad de trazabilidad en el tiempo utilizando los timestamps basta con borrarlo totalmente mediante el comando:

```
deleteall 'twitter_followers', '1,4'
```

4. **Listar usuarios que sigue el usuario X:**

```
scan 'twitter_followers', {FILTER => "PrefixFilter ('1,')"}
```

```
hbase(main):018:0> scan 'twitter_followers', {FILTER => "PrefixFilter ('1,')"}
ROW                                           COLUMN+CELL
 1,2                                          column=status:date, timestamp=1474316169232, value=Mon Sep 19 15:16:09 COT 2016
 1,2                                          column=status:is_following, timestamp=1474316169241, value=True
 1,3                                          column=status:date, timestamp=1474316169249, value=Mon Sep 19 15:16:09 COT 2016
 1,3                                          column=status:is_following, timestamp=1474316169254, value=True
 1,4                                          column=status:date, timestamp=1474316169273, value=Mon Sep 19 15:16:09 COT 2016
 1,4                                          column=status:is_following, timestamp=1474316169279, value=True
3 row(s) in 0.0140 seconds
```

En la anterior consulta se listan todos los usuarios que sigue el usuario 1, nótese que en el filtro es importante utilizar la coma ',' justo después del número de usuario que se desea buscar.

5. **Determinar si un usuario X sigue al usuario Y:**

```
get 'twitter_followers', '3,1', 'status:is_following'
```

```
hbase(main):001:0> get 'twitter_followers', '3,1', 'status:is_following'
COLUMN                                        CELL
 status:is_following                          timestamp=1474316169288, value=True
1 row(s) in 0.1490 seconds
```

En la anterior consulta se verifica si el usuario 3 sigue al usuario 1, para que sea afirmativo el valor de la columna 'is_following' debe ser 'True' en caso contrario la consulta no arrojara resultados o el valor de dicha columna debe ser 'False'.

6. **Listar los usuarios que siguen al usuario X:**

```
scan 'twitter_followers', {COLUMN => 'status:is_following', FILTER => "(RowFilter (=,'regexstring:.*,2') AND ValueFilter( =, 'binary:True'))"}
```

```
hbase(main):003:0> scan 'twitter_followers', {COLUMN => 'status:is_following', FILTER => "(RowFilter (=,'regexstring:.*,2'))"}
ROW                                                  COLUMN+CELL
 1,2                                                 column=status:is_following, timestamp=1474316169241, value=True
 4,2                                                 column=status:is_following, timestamp=1474321315836, value=True
2 row(s) in 0.0080 seconds
```

En el ejemplo anterior se listan todos los usuarios que siguen al usuario 2.