Работа с HBASE

Освежим память

- BigTable
- HBASE
- RowKey
- Master Server
- Region Server
- Hot Region
- Column family
- Column
- Happy Base
- Bulk Load

Задача №1:
Архитектура хранилища социальной сети

Пользователь

- ФИО
- Пол, возраст
- Аватар
- Интересы(много)
- Сообщества
- Друзья
- Посты
  - Комменты к постам
- Лайки

Сообщество

- Члены сообщества
- Аватар
- Посты
- Комментарии

Column families & columns

Таблица users

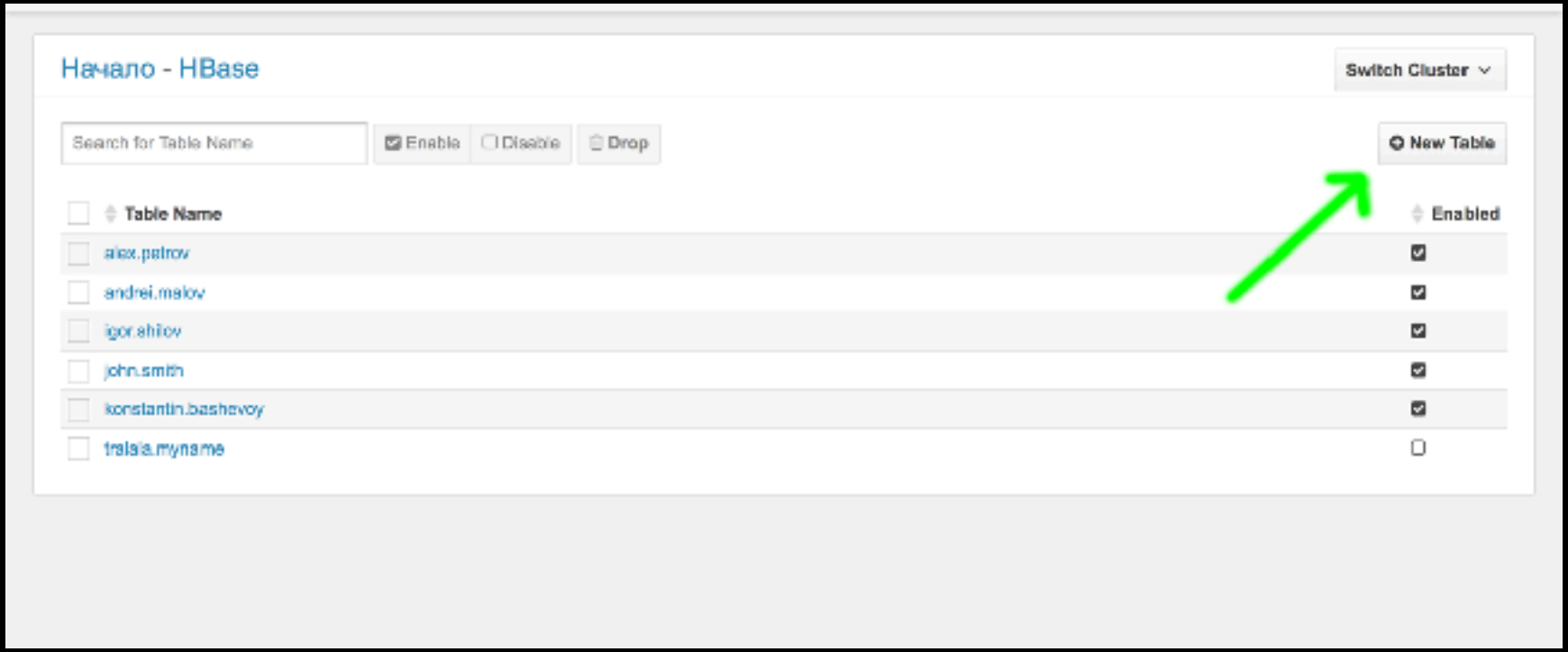
- cf:info
  - Name
  - LastName
  - …
- cf: communities
  - Community
- Cf:posts
  - Post
  - Comment1
  - Comment2
  - …
  - CommentN
- Cf: friends
  - friend

Таблица communities

- Cf:info
  - name
  - avatar
  - …
- Cf:members
  - member
- Cf:posts
  - Post
  - Comment1
  - Comment2
  - …
  - CommentN

Задача №2
Завести таблицу users

Создание таблицы через UI



Задача №3
залить data_small данные в таблицу

```
import happybase

connection =
happybase.Connection('cluster2.newprolab.com')
table=connection.table('alex.petrov’)
table.put('123123', {'info:name': 'alex',
'info:last_name': 'petrov'})
```

HappyBase

```
import happybase

connection =
happybase.Connection('cluster2.newprolab.com')
table=connection.table('alex.petrov')
data_file = open(”data_small.json")

for line in data_file:
        row = json.loads(line)
        row_key = row[‘uid’]
        table.put(row_key,row)
```

Задача №4
batch load

- Отправлять изменения пакетами по 1000 штук

- batch = table.batch()
- batch.put(row_key, row)
- batch.send()

```
import happybase
Import json

connection = happybase.Connection('hp-master')
table=connection.table('alex.petrov')
data_file = open(”data.json")
BATCH_CNT = 1000

cnt = 0
batch = table.batch()
for line in data_file:
   row = json.loads(line)
   row_key = row[‘uid’]
   batch.put(row_key, row)
   cnt += 1
   if cnt % BATCH_CNT == 0:
      batch.send()
      batch = table.batch()
      print str(cnt) + " batch sent"

batch.send()
```

Задача №5
Загрузка на mapreduce

- Читаем из stdin а не из файла
- hadoop jar /opt/cloudera/parcels/CDH-5.4.0-1.cdh5.4.0.p0.27/lib/hadoop-mapreduce/hadoop-streaming.jar -input 'info.csv' -output 'info_result' -file task2.py -mapper "python task2.py"

Bulk load

- hbase org.apache.hadoop.hbase.mapreduce.ImportTsv -Dimporttsv.columns=HBASE_ROW_KEY,info:first_name,info:last_name,info:bdate,info:age,info:gender,info:university -Dimporttsv.bulk.output=/user/john.smith/bulk_load_files alex.petrov /user/john.smith/input

- hbase org.apache.hadoop.hbase.mapreduce.LoadIncrementalHFiles bulk_load_files alex.petrov