

WQD7007 Big Data Management

Big Data Tools

HBase

HBase

- An **open source NoSQL database** that provides random and real-time read/write (CRUD operation) access to those large datasets that runs on top of HDFS.
- HBase **scales linearly to handle huge data sets with billions of rows and millions of columns**, and it easily combines data sources that use a wide variety of different structures and schemas.
 - making it a great choice to store multi-structured or sparse data.

HBase

- These following characteristics make HBase a great choice for **storing semi-structured data like log data** and then providing that data very quickly to users or applications integrated with HBase.
- Linear scaling of Hbase:
 - Require all tables to have a primary key.
 - The key space is divided into sequential blocks that are then allotted to a region.
 - RegionServers own one or more regions, so the load is spread uniformly across the cluster.

HBase

Characteristics	Benefit
Fault tolerant	<ul style="list-style-type: none">• Replication across the data center• Atomic and strongly consistent row-level operations• High availability through automatic failover• Automatic sharding and load balancings of tables
Fast	<ul style="list-style-type: none">• Near real time lookups• In-memory caching via block cache and bloom filters• Server side processing via filters and co-processors
Usable	<ul style="list-style-type: none">• Data model accommodates wide range of use cases• Metrics exports via File and Ganglia plugins• Easy Java API as well as Thrift and REST gateway APIs

HBase

- Enterprises use Apache HBase's **low latency storage** for scenarios that require real-time analysis and tabular data for end user applications.
 - Example 1: One company that provides **web security services** maintains a system accepting billions of event traces and activity logs from its customer's desktops every day.
 - The company's programmers can tightly integrate their **security solutions** with HBase (to assure that the protection they provide keeps pace with **real-time changes** in the threat landscape.)

HBase

- Example 2: One company provides **stock market ticker plant data** that its **users query more than thirty thousand times per second**, with an SLA of only a few milliseconds.
 - Apache HBase provides that **super low-latency** access over an enormous, rapidly changing data store.

HBase

- Apache HBase provides high availability in several ways:
 - **Highly available cluster topology information** through production deployments with multiple HMaster and ZooKeeper instances
 - **Data distribution across many nodes** means that loss of a single node only affects data stored on that node

HBase

- Apache HBase provides high availability in several ways:
 - **HBase HA** allows data storage, ensuring that loss of a single node does not result in loss of data availability
 - **HFile format stores data directly in HDFS.** HFile can be read or written to by Apache Hive, Apache Pig, MapReduce, and Apache Tez, permitting deep analytics on HBase without data movement

Version command

```
hbase(main):001:0> version
```

```
hbase(main):001:0> version  
1.1.2.2.5.0.0-1245, r53538b8ab6749cbb6fdc0fe448b89aa82495fb3f, Fri Aug 26 01:32:27 UTC 2016
```

list command

```
hbase(main):002:0> list
```

```
[hbase(main):002:0> list  
TABLE  
ATLAS_ENTITY_AUDIT_EVENTS  
Contacts  
atlas_titan  
iemployee  
test  
5 row(s) in 0.6480 seconds  
  
=> ["ATLAS_ENTITY_AUDIT_EVENTS", "Contacts", "atlas_titan", "iemployee", "test"]
```

Create table

```
hbase(main):003:0> create 'customer','address','order'
```

```
hbase(main):004:0> list
```

```
[hbase(main):003:0> create 'customer','address','order']  
0 row(s) in 5.5740 seconds
```

```
=> Hbase::Table - customer
```

```
[hbase(main):004:0> list
```

```
TABLE
```

```
ATLAS_ENTITY_AUDIT_EVENTS
```

```
Contacts
```

```
atlas_titan
```

```
customer
```

```
iemployee
```

```
test
```

```
6 row(s) in 0.0270 seconds
```

```
=> ["ATLAS_ENTITY_AUDIT_EVENTS", "Contacts", "atlas_titan", "customer", "iemployee", "test"]
```

Insert entry

- `hbase(main):026:0> put 'customer','john','address:city','Boston'`
 - customer is the table name
 - John is the row key
 - address is the column family
 - Boston is its value.

```
[hbase(main):005:0> put 'customer','john','address:city','Boston'  
0 row(s) in 0.4760 seconds
```

```
[hbase(main):006:0> put 'customer','john','address:state','Massachusetts'  
0 row(s) in 0.0320 seconds
```

```
[hbase(main):007:0> put 'customer','john','address:street','street1'  
0 row(s) in 0.0470 seconds
```

```
[hbase(main):008:0> put 'customer','john','order:number','ORD-15'  
0 row(s) in 0.0140 seconds
```

```
[hbase(main):009:0> put 'customer','john','order:amount','15'  
0 row(s) in 0.0860 seconds
```

Put another record

```
[hbase(main):010:0> put 'customer','Finch','address:city','Newyork'
0 row(s) in 0.0200 seconds

[hbase(main):011:0> put 'customer','Finch','address:state','Newyork'
0 row(s) in 0.0430 seconds

[hbase(main):012:0> put 'customer','Finch','order:number','ORD-16'
0 row(s) in 0.0320 seconds

[hbase(main):013:0> put 'customer','Finch','order:amount','15'
0 row(s) in 0.0190 seconds
```

Get record

hbase(main):026:0> Get 'customer', 'john'

```
[hbase(main):014:0> get 'customer','john']
COLUMN                                CELL
address:city                          timestamp=1523930451244, value=Boston
address:state                         timestamp=1523930461470, value=Massachusetts
address:street                        timestamp=1523930468611, value=street1
order:amount                         timestamp=1523930484954, value=15
order:number                         timestamp=1523930476471, value=ORD-15
5 row(s) in 0.1710 seconds
```

Get record (2)

- Using get command to retrieve the address of john
 - `hbase(main):044:0> get 'customer','john','address'`

```
[hbase(main):015:0> get 'customer','john','address']
COLUMN                                CELL
address:city                          timestamp=1523930451244, value=Boston
address:state                          timestamp=1523930461470, value=Massachusetts
address:street                         timestamp=1523930468611, value=street1
3 row(s) in 0.0420 seconds
```

Get record (3)

- Using get command to retrieve city of john
 - `hbase(main):045:0> get 'customer','john','address:city'`

```
[hbase(main):016:0> get 'customer','john','address:city'  
COLUMN                                CELL  
address:city                          timestamp=1523930451244, value=Boston  
1 row(s) in 0.0930 seconds
```


Scan record

```
hbase(main):017:0> scan customer
```

```
hbase(main):018:0> scan 'customer'
ROW                                COLUMN+CELL
Finch                             column=address:city, timestamp=1523930561402, value=Newyork
Finch                             column=address:state, timestamp=1523930568997, value=Newyork
Finch                             column=order:amount, timestamp=1523930587114, value=15
Finch                             column=order:number, timestamp=1523930576473, value=ORD-16
john                              column=address:city, timestamp=1523930451244, value=Boston
john                              column=address:state, timestamp=1523930461470, value=Massachusetts
john                              column=address:street, timestamp=1523930468611, value=street1
john                              column=order:amount, timestamp=1523930484954, value=15
john                              column=order:number, timestamp=1523930476471, value=ORD-15
2 row(s) in 0.1890 seconds
```

Update record

```
hbase(main):011:0>
```

```
put 'customer', 'john', 'address:street', 'street2'
```

```
[hbase(main):010:0> scan 'customer'
ROW                                COLUMN+CELL
 john                               column=address:street, timestamp=1523930468611, value=street1
 john                               column=order:amount, timestamp=1523930484954, value=15
 john                               column=order:number, timestamp=1523930476471, value=ORD-15
1 row(s) in 0.0520 seconds

[hbase(main):011:0> put 'customer', 'john', 'address:street', 'street2'
0 row(s) in 0.2130 seconds

[hbase(main):012:0> scan 'customer'
ROW                                COLUMN+CELL
 john                               column=address:street, timestamp=1523932537860, value=street2
 john                               column=order:amount, timestamp=1523930484954, value=15
 john                               column=order:number, timestamp=1523930476471, value=ORD-15
1 row(s) in 0.0570 seconds
```

Delete entire record

```
hbase(main):019:0> deleteall 'customer','Finch'
```

```
hbase(main):003:0> scan 'customer'
ROW                                COLUMN+CELL
Finch                             column=address:state, timestamp=1523930568997, value=Newyork
Finch                             column=order:amount, timestamp=1523930587114, value=15
Finch                             column=order:number, timestamp=1523930576473, value=ORD-16
john                              column=address:state, timestamp=1523930461470, value=Massachusetts
john                              column=address:street, timestamp=1523930468611, value=street1
john                              column=order:amount, timestamp=1523930484954, value=15
john                              column=order:number, timestamp=1523930476471, value=ORD-15
2 row(s) in 0.2870 seconds

hbase(main):004:0> deleteall 'customer','Finch'
0 row(s) in 0.0860 seconds

hbase(main):005:0> scan 'customer'
ROW                                COLUMN+CELL
john                              column=address:state, timestamp=1523930461470, value=Massachusetts
john                              column=address:street, timestamp=1523930468611, value=street1
john                              column=order:amount, timestamp=1523930484954, value=15
john                              column=order:number, timestamp=1523930476471, value=ORD-15
1 row(s) in 0.0340 seconds
```

Delete specific column

- `hbase(main):046:0> delete 'customer',john,'address:state'`

```
[hbase(main):008:0> scan 'customer'
ROW                                COLUMN+CELL
john                               column=address:state, timestamp=1523930461470, value=Massachusetts
john                               column=address:street, timestamp=1523930468611, value=street1
john                               column=order:amount, timestamp=1523930484954, value=15
john                               column=order:number, timestamp=1523930476471, value=ORD-15
1 row(s) in 0.1670 seconds

[hbase(main):009:0> delete 'customer', 'john', 'address:state'
0 row(s) in 0.0320 seconds

[hbase(main):010:0> scan 'customer'
ROW                                COLUMN+CELL
john                               column=address:street, timestamp=1523930468611, value=street1
john                               column=order:amount, timestamp=1523930484954, value=15
john                               column=order:number, timestamp=1523930476471, value=ORD-15
1 row(s) in 0.0520 seconds
```

Other commands

- count 'customer'
- disable 'customer'
- drop 'customer'