# Session 11:
# ADVANCE HBASE
# Assignment 1

**Task 1**

Explain the below concepts with an example in brief.

- Nosql Databases
- Types of Nosql Databases
- CAP Theorem
- HBase Architecture
- HBase vs RDBMS

NoSQL Databases:

NoSQL is an approach to database design that can accommodate a different variety of data models. It includes key-value pair, columnar, document and graph data models. Not Only SQL (NoSQL) is an alternate to the traditional relation databases in which data are represented in rows and columns. Traditional RDBMS are designed in such a way that the focus will be on the consistency of data and expects more constraints in data.

As RDBMS have increasingly become not suitable to meet performance, scalability and flexibility that are needed for data intensive applications, NoSQL databases have been adopted to mainstream enterprise applications. Moreover, NoSQL database is well suited to store unstructured data, which is growing rapidly compared to structure data. RDBMS does not meet the requirements of storing unstructured data like chat messages, user session data, and application logs.

Types of NoSQL databases:

In NoSQL, data are stored in different ways. It depends on the type of data getting stored. There are 4 types of NoSQL databases,

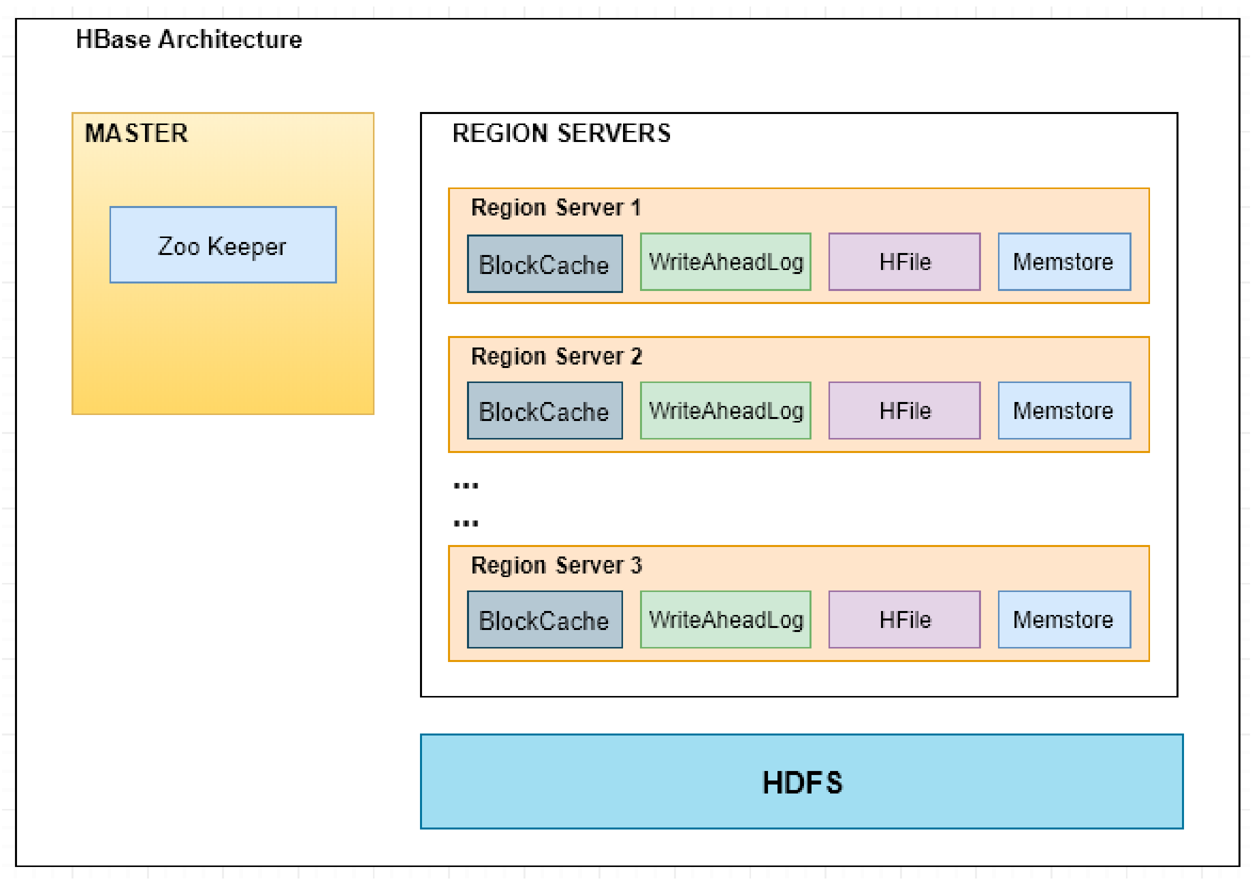
- Columnar – suitable for processing structured data. In columnar database table, the data is stored along with row id and column instead of rows. This type of storage eliminates the strict consistency in table there-by brings in dynamic.
  **Ex:** HBase and Casandra
- Document – suitable for processing semi-structured data. In this type, data is stored in key value pairs.
  **Ex:** MongoDB where data is stored in json format
- Memory – suitable for processing data in temporary distributed memory in real time. **Ex:** RedisDB
- Graph – suitable for processing and representing data in graphical format. It is used to store information about network of data.
  **Ex:** Neo4J

CAP theorem:

Generally when data systems are designed, developers or database administrators consider certain factors to meet the requirements of their application. The main factors that influence data system design are consistency, accessibility and partition tolerance. In case of distributed data systems, one has to do tradeoffs to meet their requirements. CAP theorem best explains the need of each factor and helps to understand what tradeoff one has to consider while designing data systems.

- **Consistency** – It is a guarantee that every node in a distributed cluster returns the same result. Consistency ensures every client views the same result of data.
- **Availability** – Every active node returns response for all read and write requests in a reasonable amount of time. To be available, every node on the cluster must be able to respond in a reasonable amount of time.
- **Partition Tolerant** – The system continues to function and ensures its consistency in spite of network partitions. In the world of distributed systems, data resides in partitions so that though there is a failure in one partition, other partitions can able to generate result. Partition tolerance will be impacted only when there is a total network failure which is very rare in enterprise scenarios.

HBase Architecture:



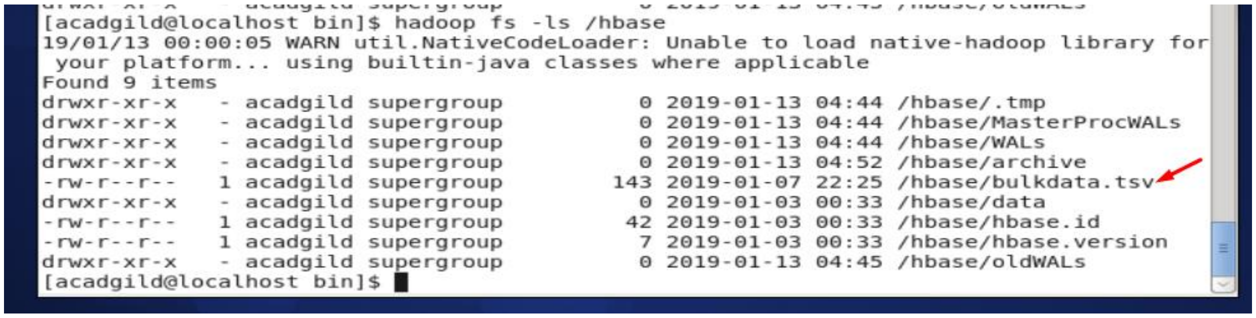
HBase vs RDBMS:

| HBASE | RDBMS |
|---|---|
| It is column oriented database. That is, each cell of data forms a meaningful data represented by rowid and column name. | It is row oriented. That is, each row of data forms a meaningful data. |
| HBase has flexible schema type | RDBMS has fixed schema |
| Supports scale out. When more memory processing power or more disk memory is required, we can just add new servers to the cluster. | Support scale up. When more memory processing power is required, we have to upgrade the existing server. |
| The amount of data depends on number of machines (nodes) in the cluster. Not dependent on one node. | It is completely dependent on the configuration of a particular server. |
| Suited for both structured and semi-structured data | Suited only for structured data |

**Task 2**

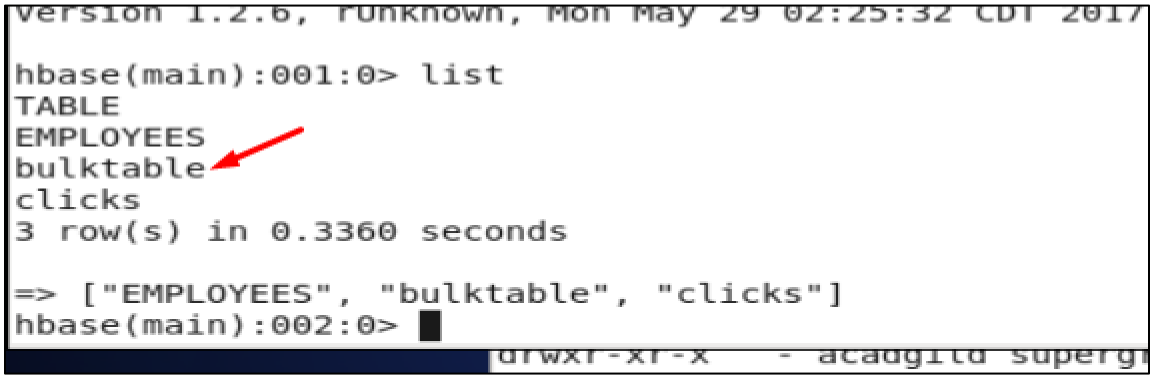
Execute blog present in below link
https://acadgild.com/blog/importtsv-data-from-hdfs-into-hbase/

Created a file called “bulktable.tsv” and moved the file from local file system to HDFS under /hbase folder,



```
[acadgild@localhost bin]$ hadoop fs -ls /hbase
19/01/13 00:00:05 WARN util.NativeCodeLoader: Unable to load native-hadoop library for
 your platform... using builtin-java classes where applicable
Found 9 items
drwxr-xr-x   - acadgild supergroup          0 2019-01-13 04:44 /hbase/.tmp
drwxr-xr-x   - acadgild supergroup          0 2019-01-13 04:44 /hbase/MasterProcWALs
drwxr-xr-x   - acadgild supergroup          0 2019-01-13 04:44 /hbase/WALs
drwxr-xr-x   - acadgild supergroup          0 2019-01-13 04:52 /hbase/archive
-rw-r--r--   1 acadgild supergroup        143 2019-01-07 22:25 /hbase/bulkdata.tsv
drwxr-xr-x   - acadgild supergroup          0 2019-01-03 00:33 /hbase/data
-rw-r--r--   1 acadgild supergroup         42 2019-01-03 00:33 /hbase/hbase.id
-rw-r--r--   1 acadgild supergroup          7 2019-01-03 00:33 /hbase/hbase.version
drwxr-xr-x   - acadgild supergroup          0 2019-01-13 04:45 /hbase/oldWALs
[acadgild@localhost bin]$
```

In HBase, created a new table called “bulktable” via Hbase shell,

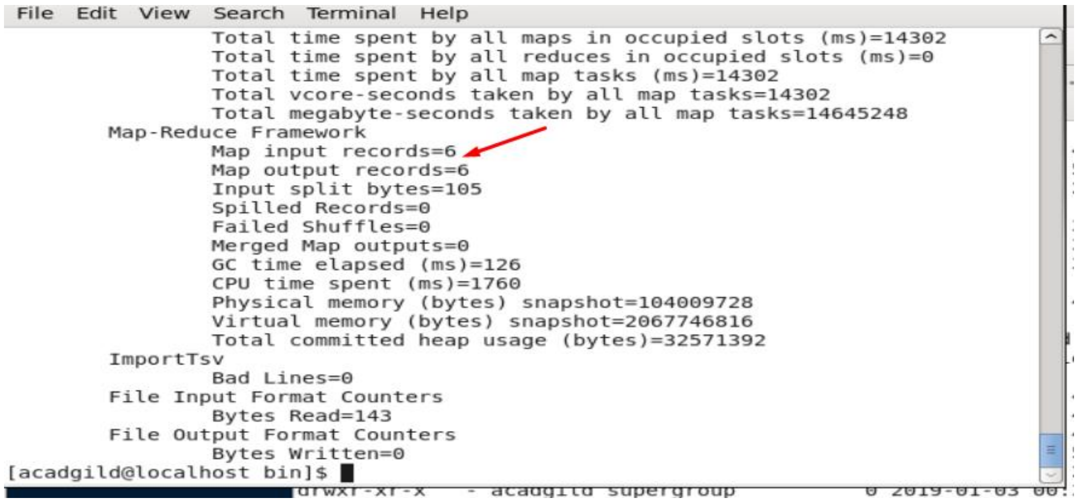


```
version 1.2.6, rUnknown, Mon May 29 02:25:32 CDT 2017

hbase(main):001:0> list
TABLE
EMPLOYEES
bulktable
clicks
3 row(s) in 0.3360 seconds

=> ["EMPLOYEES", "bulktable", "clicks"]
hbase(main):002:0>
```

Execute the command below in HBase bin folder path, to import data from bulkdata.tsv to bulktable table,

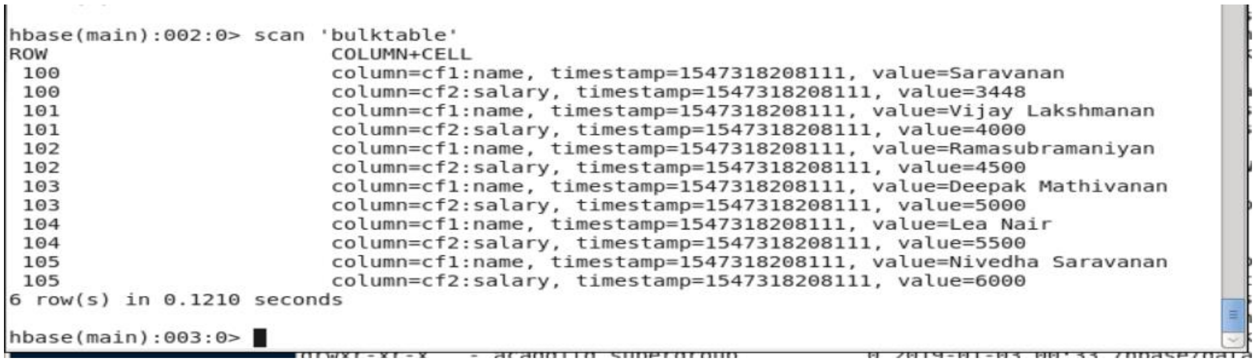
*hbase org.apache.hadoop.hbase.mapreduce.ImportTsv -Dimporttsv.columns="HBASE_ROW_KEY,cf1:name,cf2:salary" bulktable /hbase/bulkdata.tsv*

The job is submitted and the execution is completed with below message,



```
                Total time spent by all maps in occupied slots (ms)=14302
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=14302
                Total vcore-seconds taken by all map tasks=14302
                Total megabyte-seconds taken by all map tasks=14645248
        Map-Reduce Framework
                Map input records=6
                Map output records=6
                Input split bytes=105
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=126
                CPU time spent (ms)=1760
                Physical memory (bytes) snapshot=104009728
                Virtual memory (bytes) snapshot=2067746816
                Total committed heap usage (bytes)=32571392
        ImportTsv
                Bad Lines=0
        File Input Format Counters
                Bytes Read=143
        File Output Format Counters
                Bytes Written=0
[acadgild@localhost bin]$
```

Then scanned the table ‘bulktable’ to confirm the data import,



```
hbase(main):002:0> scan 'bulktable'
ROW                         COLUMN+CELL
 100                        column=cf1:name, timestamp=1547318208111, value=Saravanan
 100                        column=cf2:salary, timestamp=1547318208111, value=3448
 101                        column=cf1:name, timestamp=1547318208111, value=Vijay Lakshmanan
 101                        column=cf2:salary, timestamp=1547318208111, value=4000
 102                        column=cf1:name, timestamp=1547318208111, value=Ramasubramaniyan
 102                        column=cf2:salary, timestamp=1547318208111, value=4500
 103                        column=cf1:name, timestamp=1547318208111, value=Deepak Mathivanan
 103                        column=cf2:salary, timestamp=1547318208111, value=5000
 104                        column=cf1:name, timestamp=1547318208111, value=Lea Nair
 104                        column=cf2:salary, timestamp=1547318208111, value=5500
 105                        column=cf1:name, timestamp=1547318208111, value=Nivedha Saravanan
 105                        column=cf2:salary, timestamp=1547318208111, value=6000
6 row(s) in 0.1210 seconds

hbase(main):003:0>
```

The data in ‘bulkdata.tsv’ have been successfully imported in to ‘bulktable’ HBase table.