Vijay's Assignment – Hbase2

Task1

Explain the below concepts with an example in brief.

● Nosql Databases

NoSQL DB are non relational databases provide mechanism to store and retireve data that is different from the traditional tabular format used by RDBMS. NoSQL is designed to cater to the fast expanding structured, unstructured and hybrid data from various sources like mobile, social media etc that cannot be stored in RDBMS efficiently. NoSQL DB are designed to be implemented in cheap low cost commoditiy hardware like cloud.

Example: Mongo DB, Cassandra, Hbase etc

● Types of Nosql Databases

a) Key-value: Data is stored as a collection of key and value pairs Ex: Aerospike, Redis etc
b) Document store: Data is stored in some standard formats like XML, JSON etc. Documents are addressed with a unique key within the DB ex: MongoDB, CouchDB
c) Graph: Stores data which are interconnected among themselves with finite relations between them Ex: IBM Graph, Neo4j
d) Wide colum: Stores columns of data together instead of rows Ex: Hbase, Cassandra

● CAP Theorem

CAP stands for Consistency, Availability & Partition tolerance. This theorum is very critical in the Big Data world especially when we need to make trade off's between the three based on the use case.

Partition Tolerance: - Keeping the system running inspite of any amount of network failures. Partition tolerant systems can sustain any number of network failures as long as the entire network is not down. Data records are sufficiently replicated across the networks and nodes to keep the system up during intermittent outages.

High Consistency: - All nodes see the same data at the same time. Any read operation from any node will retrieve only the latest data

High Availability: - Every request gets a success/failure response across all the nodes in the network

It is not possibile for any DB to achieve all these three requirements at the same time. Each DB tries to achieve two of these factors and they can be applied based on the use case.

Consistency & availability – RDBMS, Aster Data

Availability & Partition tolerance – Dynamo DB, Cassandra, Couch DB etc

Consistency & Partition Tolerance – Hbase, Mongo DB, Redis

● HBase Architecture

HBasd follows the Master Slave architecture. There are four major components in HBase architecture

- HMaster
- HRegionserver
- HRegions
- Zookeeper

HMaster

1) This is the master server
2) Maintains the metadata about region servers present in the cluster
3) Interface for all metadata changes
4) Runs on the Namenode
5) Assigns regions to region servers
6) Handles load balacing and fail over operations
7) Schema and meta data change is handled by HMaster

HRegionserver

1) This is the region server implementation
2) Serves and manages the regions present in the cluster
3) Runs on Datanode
4) Handles read & write requests and communicates with client directly

HRegions

1) Place where the actual data is stored. Contains the distribution of tables and column faimilies
2) Each column family has a separate store
3) Contains two major components
    a. Memstore – Physical RAM
    b. HFile – actual Hadoop file where data is stored

Zookeeper

1) Centralized monitoring server that maintains config information and provides distributed synchronization
2) Client communication establishment with region servers
3) Tracks network failures and network partitions

● HBase vs RDBMS

Hbase	RDBMS
Column oriented	Row oriented
Flexible schema, columns are created on the fly	Fixed schema
Good with sparse tables	Not optimzed for sparsed tables
No Query Language	SQL Support
Wide tables	Narrow tables
Non optimized joins – uses MapReduce	Optimized joins (small & faster)

Denormalize your data	Normalize as you can
Horizontal scalable by adding new Hardware	Not easy to scale
Not suitable for transactional processing	Best for transactional processing
Works well with semi structured and unstructured data	Good for structured data only

Task2

Execute blog present in below link

https://acadgild.com/blog/importtsv-data-from-hdfs-into-hbase/

“bulktable” created in HBase

```
hbase(main):014:0> create 'bulktable','cf1','cf2'
0 row(s) in 1.2610 seconds
```

TSV file bulk_data.tsv created in local FS and copied over to HDFS

```
[acadgild@localhost Hbase]$ ls
bulk_data.tsv
[acadgild@localhost Hbase]$ hadoop fs -cat /HBASE/bulk_data.tsv
19/01/09 19:25:03 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
1       VIJAY   36
2       DIVYA   33
3       BHUVAN  2
4       RAMASUBRAMANIAN 68
5       GOWRI   61
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost Hbase]$ hbase org.apache.hadoop.hbase.mapreduce.ImportTsv -D
importtsv.columns=HBASE_ROW_KEY,cf1:name,cf2:age bulktable /HBASE/bulk_data.tsv
```

Import completed

```
        importtsv
                Bad Lines=0
        File Input Format Counters
                Bytes Read=65
        File Output Format Counters
                Bytes Written=0
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost Hbase]$
```

“bulkdata” tabled scanned in HBase

```
hbase(main):019:0> scan 'bulktable'
ROW                     COLUMN+CELL
 1                      column=cf1:name, timestamp=1547042186084, value=VIJAY
 1                      column=cf2:age, timestamp=1547042186084, value=36
 2                      column=cf1:name, timestamp=1547042186084, value=DIVYA
 2                      column=cf2:age, timestamp=1547042186084, value=33
 3                      column=cf1:name, timestamp=1547042186084, value=BHUVAN
 3                      column=cf2:age, timestamp=1547042186084, value=2
 4                      column=cf1:name, timestamp=1547042186084, value=RAMASUBRAM
                        ANIAN
 4                      column=cf2:age, timestamp=1547042186084, value=68
 5                      column=cf1:name, timestamp=1547042186084, value=GOWRI
 5                      column=cf2:age, timestamp=1547042186084, value=61
5 row(s) in 0.1270 seconds
```