



SNS COLLEGE OF TECHNOLOGY



(An Autonomous Institution)

Re-accredited by NAAC with A+ grade, Accredited by NBA(CSE, IT, ECE, EEE & Mechanical)
Approved by AICTE, New Delhi, Recognized by UGC, Affiliated to Anna University, Chennai

Department of MCA

Topic: **HBase**

Course

19CAT702
Big Data Analytics

Unit IV

Frameworks

Elective

III Semester /
II MCA



Session Objectives



- Differentiate DBMS and column oriented databases
- Understand how data are presented in the Hbase data model





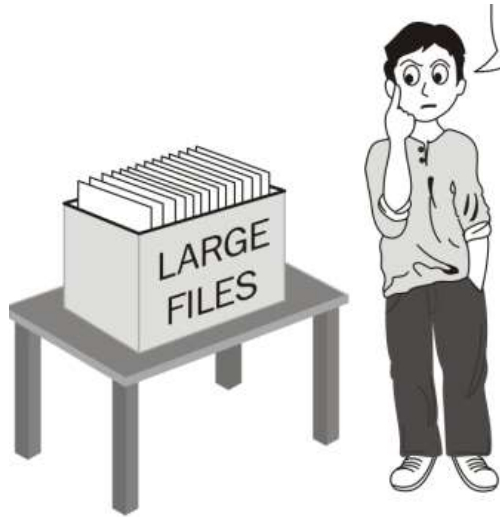
HBase - Introduction



- A Distributed column-oriented database
- Open source, non-relational, multidimensional and Scalable
- Built on HDFS
- Designed to provide a fault tolerant way of storing large collection of sparse data sets
- Hosting of very large table - billions of rows X millions of columns



HBase - Introduction



- Fast lookup for large tables
- Scale down from the opposite direction
- built from the ground-up to scale linearly just by adding nodes



HBase – Data Model



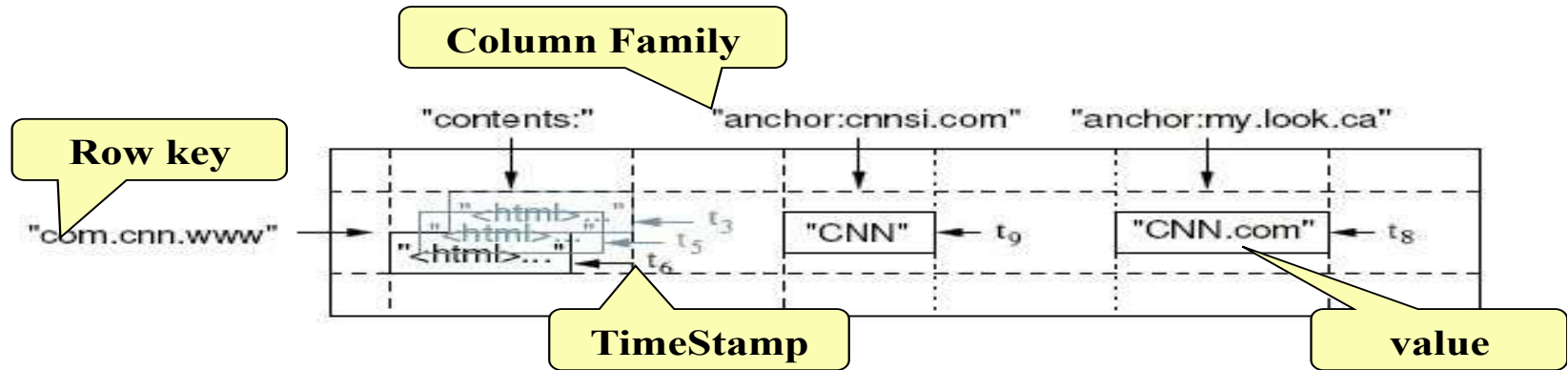
- Applications store data into labeled tables, made of rows and columns
- Table cells—are versioned , a timestamp auto-assigned by HBase at the time of cell insertion
- cell's content is an uninterpreted array of bytes
- Table row keys are also byte arrays
- Row columns are grouped into *column families*
- All column family members have a common prefix, for example *temperature:air* and *temperature:dew_point* are members of the *temperature* column family



Hbase – Row Model



- HBase is based on Google's Bigtable model
 - Key-Value pairs





HBase – Column families



Each row has a **Key**

Each record is divided into **Column Families**



Each column family consists of one or more **Columns**



HBase – Data Model



- ❑ Tables are automatically partitioned horizontally into regions.
- ❑ Each region comprises a subset of a table's rows
- ❑ Initially, a table comprises a single region, but as the size of the region grows, crosses a configurable size threshold, it splits at a row boundary into two new regions of approximately equal size
- ❑ A table that is too big for any one server can be carried by a cluster of servers with each node hosting a subset of the table's total regions.





Hbase – Row Model



- ❑ Each column family is stored in a separate file (called Htables)
- ❑ Key & Version numbers are replicated with each column family
- ❑ Empty cells are not stored

Table 5.3. ColumnFamily contents

Row Key	Time Stamp	ColumnFamily "contents:"
"com.cnn.www"	t6	contents:html = "<html>..."
"com.cnn.www"	t5	contents:html = "<html>..."
"com.cnn.www"	t3	contents:html = "<html>..."

Table 5.2. ColumnFamily anchor

Row Key	Time Stamp	Column Family anchor
"com.cnn.www"	t9	anchor:cnnsi.com = "CNN"
"com.cnn.www"	t8	anchor:my.look.ca = "CNN.com"

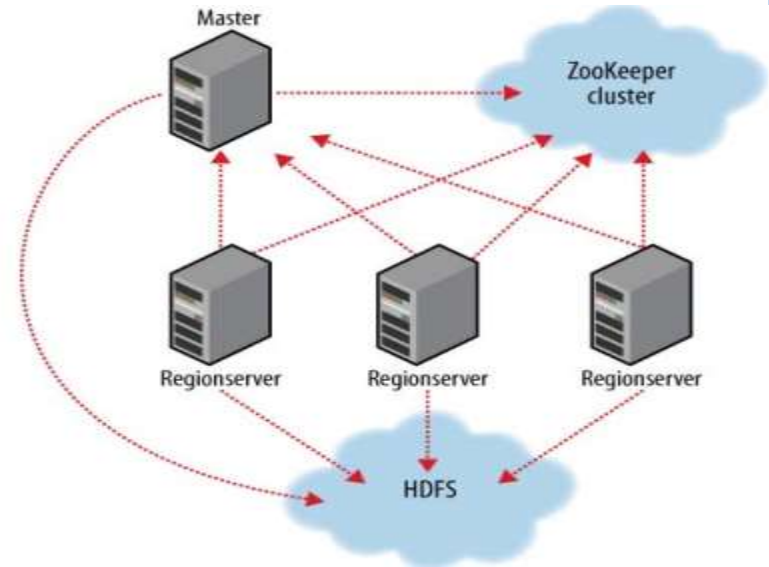
HBase maintains a multi-level index on values:
<key, column family, column name, timestamp>



HBase – Implementation



- ❑ HBase modeled with an HBase *master* node orchestrating a cluster of one or more *regionserver* slaves
- ❑ HBase master is responsible for
 - *bootstrapping a virgin install*
 - *assigning regions to region servers*
 - *recovering regionserver failures*





HBase – Implementation



- ❑ Regionservers carry zero or more regions and field client read/write requests
- ❑ It depends on ZooKeeper and keeps ZooKeeper instance as the authority on cluster state
- ❑ Assignment of regions is mediated via ZooKeeper in case participating servers crash mid-assignment
- ❑ Regionserver slave nodes are listed in the HBase conf/regionservers file as you would list datanodes and tasktrackers in the Hadoop conf/slaves file.



Hbase – Components



- A subset of a table's rows, like horizontal range partitioning
- Automatically done

Region

- Manages data regions
- Serves data for reads and writes (using a log)

RegionServer

- Responsible for coordinating the slaves
- Assigns regions, detects failures
- Admin functions

Master



Hbase – Features



No Real Indexes



Scale linearly & automatically with new nodes



Fault tolerance



Automatic partitioning
As tables grow, it will automatically be split into regions



Commodity hardware



Batch processing-
MapReduce integration allows fully parallel, distributed jobs against your data with locality awareness



Comparison



HBase	Database
Column oriented	Row oriented
Flexible schema	Fixed schema
Good with sparse table	Not optimized for sparse tables
Wide tables	Narrow tables
Join using MR, not optimized	Optimized for Joins
No transactions	Transactional
Good for both semi structured and structured data	Good for structured data



HBase – Operations

- ❑ Operations based on row keys
- ❑ Single row-operations
 - Put
 - Get
 - Scan
- ❑ Multi-row operations
 - Scan
 - Multiput
- ❑ No built-in Joins (use MapReduce)

