Introduction to Hbase

Tushar B. Kute,
http://tusharkute.com

Limitation of Hadoop

- Hadoop can perform only batch processing, and data will be accessed only in a sequential manner.
- That means one has to search the entire dataset even for the simplest of jobs.
- A huge dataset when processed results in another huge data set, which should also be processed sequentially.
- At this point, a new solution is needed to access any point of data in a single unit of time (random access).

What is Hbase?

- HBase is a distributed column-oriented database built on top of the Hadoop file system.
- It is an open-source project and is horizontally scalable.
- HBase is a data model that is similar to Google's big table designed to provide quick random access to huge amounts of structured data.
- It leverages the fault tolerance provided by the Hadoop File System (HDFS).
- It is a part of the Hadoop ecosystem that provides random real-time read/write access to data in the Hadoop File System.

What is Hbase?

- HBase is an open-source, column-oriented distributed database system in a Hadoop environment.
- Initially, it was Google Big Table, afterward; it was renamed as HBase and is primarily written in Java. Apache HBase is needed for real-time Big Data applications.
- HBase can store massive amounts of data from terabytes to petabytes. The tables present in HBase consist of billions of rows having millions of columns.
- HBase is built for low latency operations, which is having some specific features compared to traditional relational models.

Why to choose?

- A table for a popular web application may consist of billions of rows.
- If we want to search a particular row from such a huge amount of data, HBase is the ideal choice as query fetch time is less. Most of the online analytics applications use HBase.
- Traditional relational data models fail to meet the performance requirements of very big databases.
- These performance and processing limitations can be overcome by Apache HBase.

Hbase Features

- HBase is built for low latency operations
- HBase is used extensively for random read and write operations
- HBase stores a large amount of data in terms of tables
- Provides linear and modular scalability over cluster environment
- Strictly consistent to read and write operations
- Automatic and configurable sharding of tables
- Automatic failover supports between Region Servers
- Convenient base classes for backing Hadoop MapReduce jobs in HBase tables
- Easy to use Java API for client access
- Block cache and Bloom Filters for real-time queries
- Query predicate pushes down via server-side filters.

Importance of NoSQL in Hadoop

- In big data analytics, Hadoop plays a vital role in solving typical business problems by managing large data sets and gives the best solutions in analytics domain.
- In the Hadoop ecosystem, each component plays its unique role for the
  - Data processing
  - Data validation
  - Data storing

Importance of NoSQL in Hadoop

- In terms of storing unstructured, semi-structured data storage as well as retrieval of such data's, relational databases are less useful.
- Also, fetching results by applying query on huge data sets that are stored in Hadoop storage is a challenging task.
- NoSQL storage technologies provide the best solution for faster querying on huge datasets.

Other NoSQL Storage

- Some of the NoSQL models present in the market are Cassandra, MongoDB, and CouchDB. Each of these models has different ways of storage mechanism.
- For example, MongoDB is a document-oriented database from the NoSQL family tree. Compared to traditional databases, it provides the best features in terms of performance, availability, and scalability. It is an open-source document-oriented database, and it's written in C++.
- Cassandra is also a distributed database from open-source Apache software which is designed to handle a huge amount of data stored across commodity servers. Cassandra provides high availability with no single point of failure.

How NoSQL is different?

- HBase storage model is different from other NoSQL models discussed before. This can be stated as follow.
  - HBase stores data in the form of key/value pairs in a columnar model. In this model, all the columns are grouped together as Column families.
  - HBase provides a flexible data model and low latency access to small amounts of data stored in large data sets.
  - HBase on top of Hadoop will increase the throughput and performance of distributed cluster set up. In turn, it provides faster random reads and writes operations.

Which NoSQL to use?

DataBase Type Based on Feature	Example of Database	Use case (When to Use)
Key/ Value	Redis, MemcacheDB	Caching, Queue-ing, Distributing information
Column-Oriented	Cassandra, HBase	Scaling, Keeping Unstructured, non-volatile
Document-Oriented	MongoDB, Couchbase	Nested Information, JavaScript friendly
Graph-Based	OrientDB, Neo4J	Handling Complex relational information. Modeling and Handling classification.

Hbase vs. Hive

Features	HBase	Hive
DataBase model	Wide Column store	Relational DBMS
Data Schema	Schema- free	With Schema
SQL Support	No	Yes, it uses HQL(Hive query language)
Partition methods	Sharding	Sharding
Consistency Level	Immediate Consistency	Eventual Consistency
Secondary indexes	No	Yes
Replication Methods	Selectable replication factor	Selectable replication factor

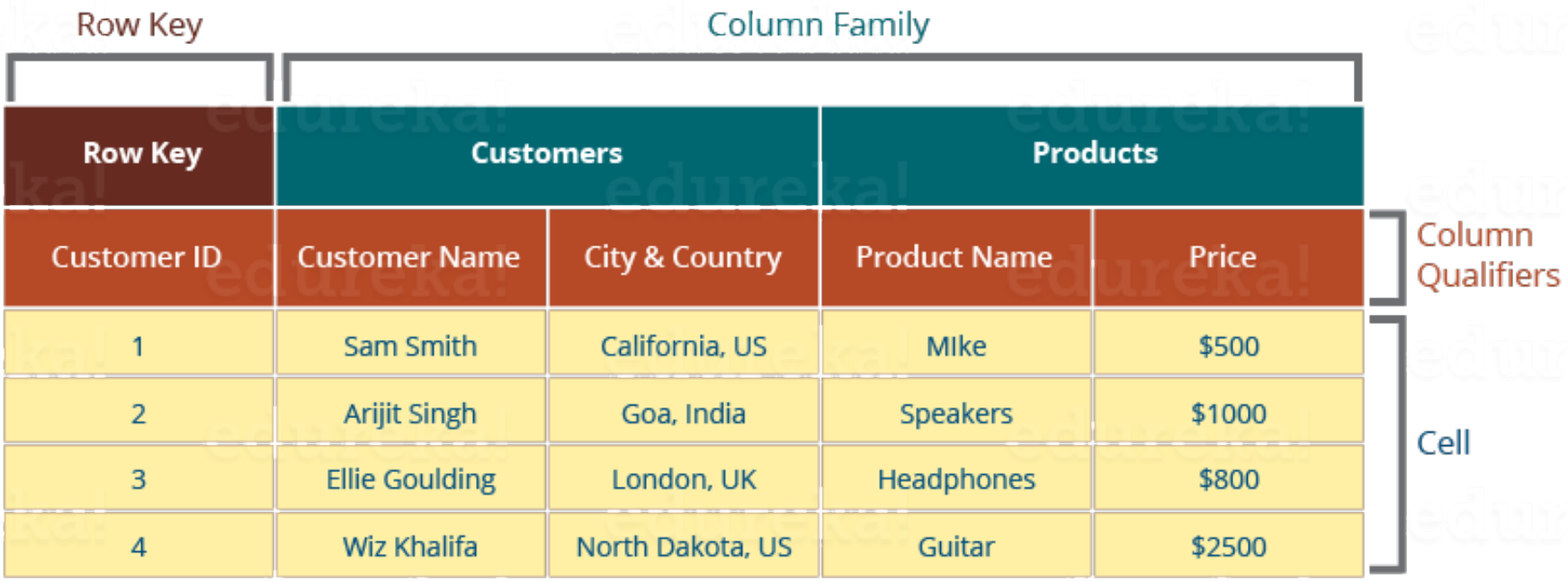
Storage Mechanism in Hbase

- HBase is a column-oriented database and the tables in it are sorted by row. The table schema defines only column families, which are the key value pairs.
- A table have multiple column families and each column family can have any number of columns.
- Subsequent column values are stored contiguously on the disk.

Storage Mechanism in Hbase

- Each cell value of the table has a timestamp. In short, in an HBase:
  - Table is a collection of rows.
  - Row is a collection of column families.
  - Column family is a collection of columns.
  - Column is a collection of key value pairs.

Storage Mechanism in Hbase



Column oriented vs. Row Oriented

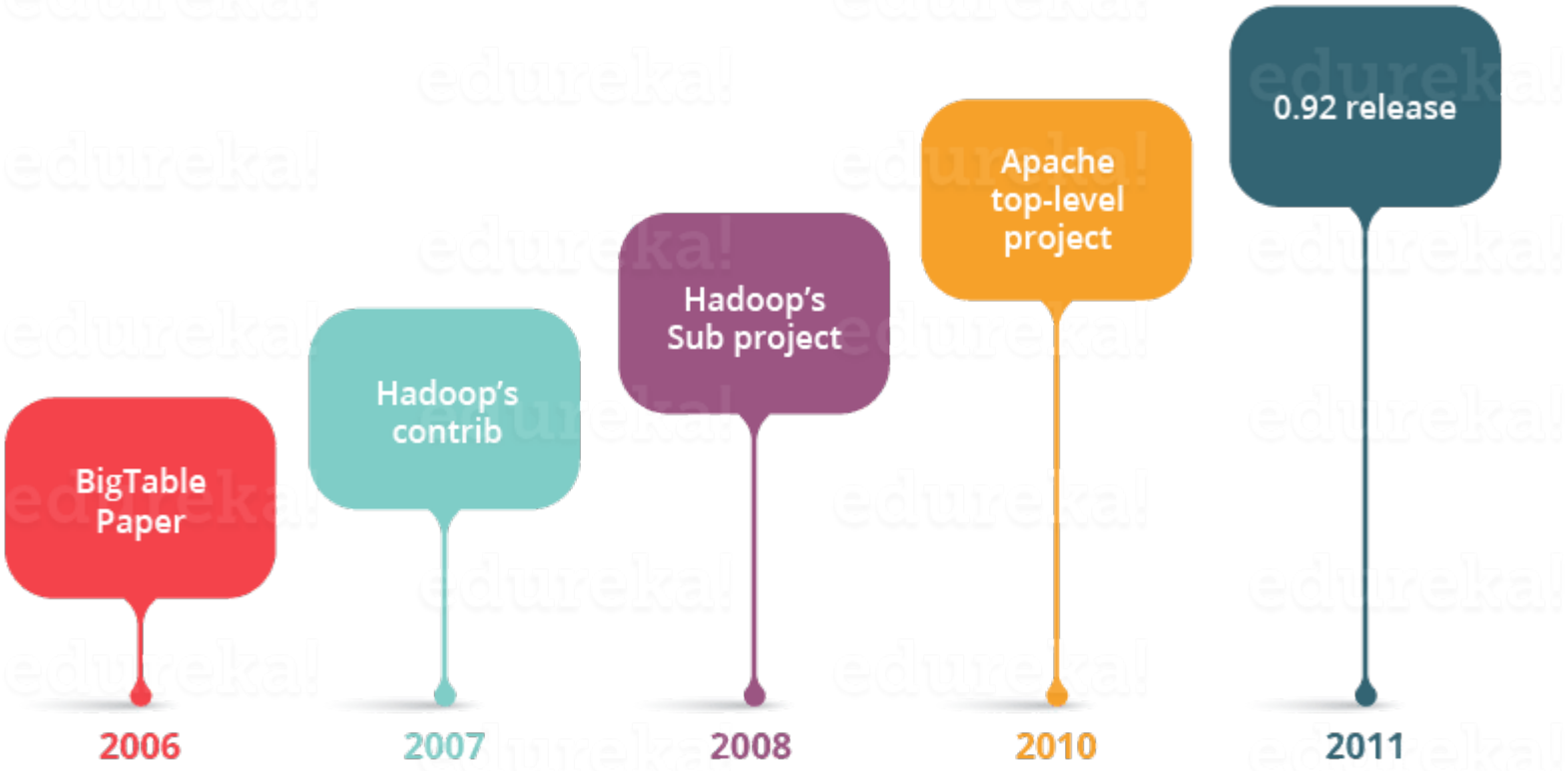
- Column-oriented databases are those that store data tables as sections of columns of data, rather than as rows of data.
- Shortly, they will have column families.

Row-Oriented Database	Column-Oriented Database
It is suitable for Online Transaction Process (OLTP).	It is suitable for Online Analytical Processing (OLAP).
Such databases are designed for small number of rows and columns.	Column-oriented databases are designed for huge tables.

Hbase vs. RDBMS

HBase	RDBMS
HBase is schema-less, it doesn't have the concept of fixed columns schema; defines only column families.	An RDBMS is governed by its schema, which describes the whole structure of tables.
It is built for wide tables. HBase is horizontally scalable.	It is thin and built for small tables. Hard to scale.
No transactions are there in HBase.	RDBMS is transactional.
It has de-normalized data.	It will have normalized data.
It is good for semi-structured as well as structured data.	It is good for structured data.

Timeline


BigTable Paper
2006
Hadoop's contrib
2007
Hadoop's Sub project
2008
Apache top-level project
2010
0.92 release
2011

Applications

- It is used whenever there is a need to write heavy applications.
- HBase is used whenever we need to provide fast random access to available data.
- Companies such as Facebook, Twitter, Yahoo, and Adobe use HBase internally.

Thank you

This presentation is created using LibreOffice Impress 5.1.6.2, can be used freely as per GNU General Public License

@mitu_skillologies

/mITuSkillologies

@mitu_group

/company/mitu-skillologies

MITUSkillologies

Web Resources
https://mitu.co.in
http://tusharkute.com

contact@mitu.co.in

tushar@tusharkute.com