CS403 - PROGRAMMING PARADIGMS

Viju P Poonthottam
Asst. Professor Dept. of Computer Science & Engineering
MES CE Kuttippuram

June 8, 2023

Table of contents

Module - 4 (Pig, Hive, Hbase)

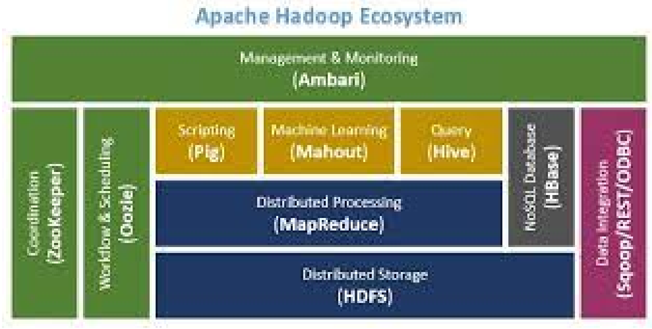
- Pig : Introduction to PIG, Execution Modes of Pig
- Comparison of Pig with Databases, Grunt.
- Pig Latin, User Defined Functions, Data Processing operators
- Hive : Hive Shell, Hive Services
- Hive Metastore, Comparison with Traditional Databases.
- HiveQL, Tables, Querying Data and User Defined Functions.
- Hbase : HBasics, Concepts, Clients, Example, Hbase Versus RDBMS.

Pig -Introduction

- Not all data analysts were familiar with Map Reduce
- Introduced by Yahoo researchers
- As an animal pig eats anything, Pig can work upon any kind of data

I Eat Anything
Pig
Data
Big Data Money

Pig Architecture


Apache Hadoop Ecosystem
Management & Monitoring (Ambari)
Coordination (ZooKeeper)
Workflow & Scheduling (Oozie)
Scripting (Pig)
Machine Learning (Mahout)
Query (Hive)
NoSQL Database (HBase)
Data Integration (Sqoop/REST/ODBC)
Distributed Processing (MapReduce)
Distributed Storage (HDFS)

Benefits of Pig

- Ease of Coding
- Optimization
- Extensibility
- Flexible
- In-built operators
- Less code
- Reusability
- Nested data types

Pig Models

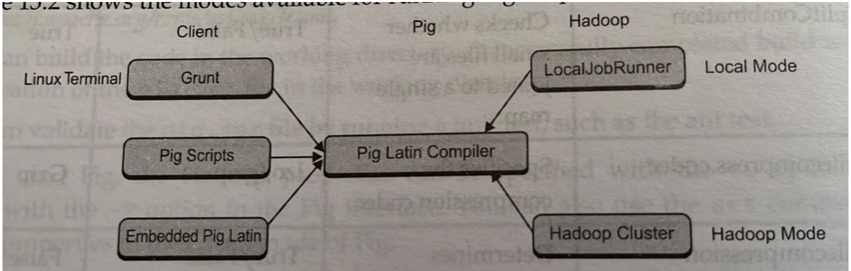


Figure 13.2: Pig Models

Pig Models

- Local mode
- Mapreduce mode

Pig Latin

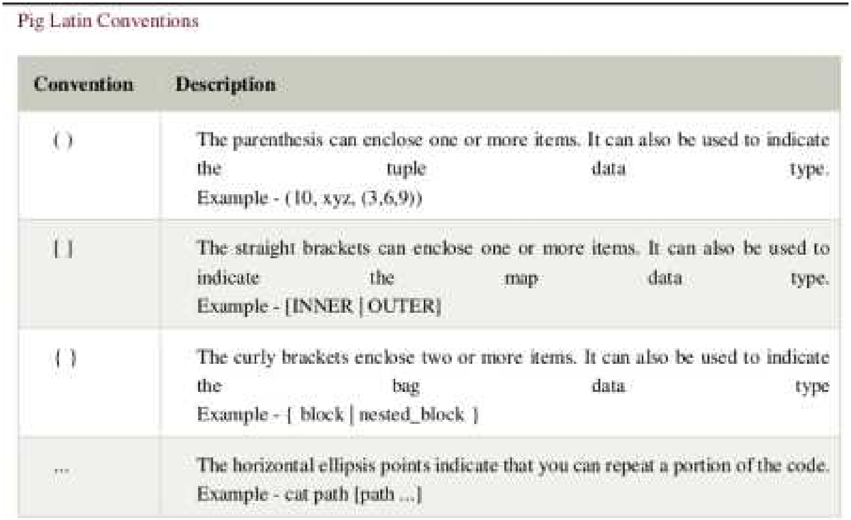
- The Pig Latin is a data flow language used by Apache Pig to analyze the data in Hadoop. It is a textual language that abstracts the programming from the Java MapReduce idiom into a notation.

Pig Latin Statements

The Pig Latin statements are used to process the data. It is an operator that accepts a relation as an input and generates another relation as an output.

- It can span multiple lines.
- Each statement must end with a semi-colon.
- It may include expression and schemas.
- By default, these statements are processed using multi-query execution

Pig Latin Conventions



Pig Latin Conventions

Convention	Description
( )	The parenthesis can enclose one or more items. It can also be used to indicate the tuple data type. Example - (10, xyz, (3,6,9))
[ ]	The straight brackets can enclose one or more items. It can also be used to indicate the map data type. Example - [INNER | OUTER]
{ }	The curly brackets enclose two or more items. It can also be used to indicate the bag data type Example - { block | nested_block }
...	The horizontal ellipsis points indicate that you can repeat a portion of the code. Example - cat path [path ...]

Latin Data Types

- int
- long
- float
- double
- chararray
- bytearray
- boolean
- datetime
- biginteger
- bigdecimal

Apache Pig Execution Mechanisms

- Interactive Mode Grunt shell
- Batch Mode Script
- Embedded Mode (UDF)

Pig Commands.

- load -Reads data from the system
- Store- Writes data to file system
- foreach -Applies expressions to each record and outputs one or more records
- filter-Applies predicate and removes records that do not return true
- Group/cogroup -Collects records with the same key from one or more inputs
- join-Joins two or more inputs based on a key
- order- Sorts records based on a key
- distinct -Removes duplicate records
- union -Merges data sets
- split -Splits data into two or more sets based on filter conditions
- stream - Sends all records through a user-provided binary
- dump -Writes output to stdout
- limit -Limits the number of records

Complex Types

- tuple - It defines an ordered set of fileds Example - *(15,12)*
- bag -It defines a collection of tuples Example - *(15,12), (12,15)*
- map -It defines a set of key value pairs Example - *[open#apache]*

Eval Functions

- AVG() - To compute the average of the numerical values within a bag.
- BagToString() -To concatenate the elements of a bag into a string. While concatenating, we can place a delimiter between these values (optional).
- CONCAT() -To concatenate two or more expressions of same type.
- COUNT() -To get the number of elements in a bag, while counting the number of tuples in a bag.
- COUNT_STAR()- It is similar to he COUNT() function. It is used to get the number of elements in a bag.
- DIFF() -To compare two bags (fields) in a tuple.

Eval Functions-conti...

- IsEmpty() -To check if a bag or map is empty.
- MAX() -To calculate the highest value for a column (numeric values or chararrays) in a single-column bag.
- MIN() -To get the minimum (lowest) value (numeric or chararray) for a certain column in a single-column bag.
- PluckTuple() -Using the Pig Latin PluckTuple() function, we can define a string Prefix and filter the columns in a relation that begin with the given prefix.

Eval Functions-conti...

- SIZE() -To compute the number of elements based on any Pig data type.
- SUBTRACT() -To subtract two bags. It takes two bags as inputs and returns a bag which contains the tuples of the first bag that are not in the second bag.
- SUM() -To get the total of the numeric values of a column in a single-column bag.
- TOKENIZE() -To split a string (which contains a group of words) in a single tuple and return a bag which contains the output of the split operation.

Hive

- Hive is a data warehouse infrastructure tool to process structured data in Hadoop. It resides on top of Hadoop to summarize Big Data, and makes querying and analyzing easy.
- It is a platform used to develop SQL type scripts to do MapReduce operations.
- Initially Hive was developed by Facebook
- open source under the name Apache Hive
- Amazon uses it in Amazon Elastic MapReduce.

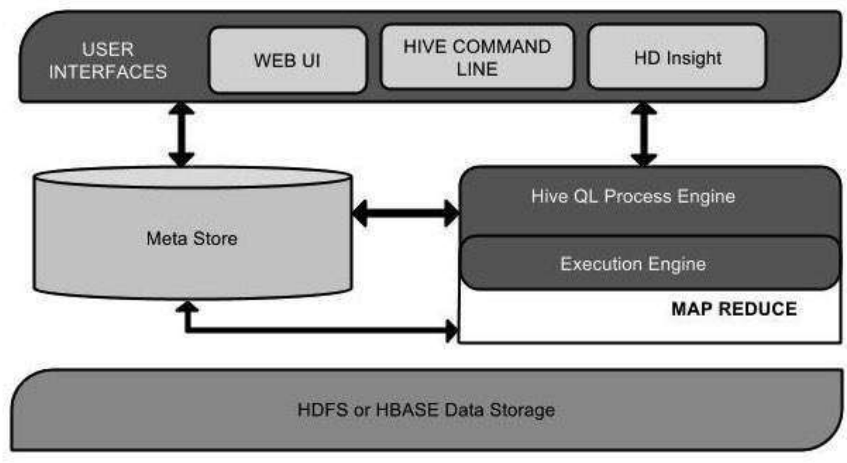
Hive is NOT

- A relational database
- A design for OnLine Transaction Processing (OLTP)
- A language for real-time queries and row-level updates

Features of Hive

- It stores schema in a database and processed data into HDFS.
- It is designed for OLAP.
- It provides SQL type language for querying called HiveQL or HQL.
- It is familiar, fast, scalable, and extensible.

Architecture of Hive



User Interface

- Hive is a data warehouse infrastructure software that can create interaction between user and HDFS. The user interfaces that Hive supports are Hive Web UI, Hive command line, and Hive HD Insight (In Windows server).

Meta Store

- Hive chooses respective database servers to store the schema or Metadata of tables, databases, columns in a table, their data types, and HDFS mapping.

HiveQL Process Engine

- HiveQL is similar to SQL for querying on schema info on the Metastore. It is one of the replacements of traditional approach for MapReduce program. Instead of writing MapReduce program in Java, we can write a query for MapReduce job and process it.

Execution Engine

- The conjunction part of HiveQL process Engine and MapReduce is Hive Execution Engine. Execution engine processes the query and generates results as same as MapReduce results. It uses the flavor of MapReduce.

HDFS or HBASE

- Hadoop distributed file system or HBASE are the data storage techniques to store data into file system.

Hive Commands

- Hive supports Data definition Language(DDL), Data Manipulation Language(DML) and User defined functions.

Hive DDL Commands

- create database
- drop database
- create table
- drop table
- alter table
- create index
- create view

Hive DML Commands

- Select
- Where
- Group By
- Order By
- Load Data
- Join:
  1. Inner Join
  2. Left Outer Join
  3. Right Outer Join
  4. Full Outer Join

Hive DDL Commands

- Create Database Statement

Hive Query Language

- The Hive Query Language (HiveQL) is a query language for Hive to process and analyze structured data in a Metastore.

Bucketing

- Bucketing concept is based on (hashing function on the bucketed column) mod (by total number of buckets). The hash_function depends on the type of the bucketing column.
- Records with the same bucketed column will always be stored in the same bucket.
- We use CLUSTERED BY clause to divide the table into buckets.
- Physically, each bucket is just a file in the table directory, and Bucket numbering is 1based.
- Bucketing can be done along with Partitioning on Hive tables and even without partitioning.

Bucketing conti...

- Bucketed tables will create almost equally distributed data file parts, unless there is skew in data.
- Bucketing is enabled by setting hive.enforce.bucketing= true;

Advantages

- Bucketed tables offer efficient sampling than by nonbucketed tables. With sampling, we can try out queries on a fraction of data for testing and debugging purpose when the original data sets are very huge.
- As the data files are equal sized parts, mapside joins will be faster on bucketed tables than non bucketed tables.
- Bucketing concept also provides the flexibility to keep the records in each bucket to be sorted by one or more columns. This makes mapside joins even more efficient, since the join of each bucket becomes an efficient mergesort.

Data Model

- Tables
- Partitions
- Buckets

Data Model -Tables

- Typed columns (int, float, string, boolean)
- Also, list: map (for JSONlike data)

Data Model -Partitions

- For example, range-partition tables by date

Data Model -Buckets

- Hash partitions within ranges (useful for sampling,join optimization)

Metastore

- Database: namespace containing a set of tables
- Holds table definitions (column types, physical layout)
- Holds partitioning information
- Can be stored in Derby, MySQL, and many other relational databases

Hive Vs RDBMS

- Latency for Hive queries is generally very high (minutes) even when data sets involved are very small
- On RDBMSs, the analyses proceed much more iteratively with the response times between iterations being less than a few minutes.
- Hive aims to provide acceptable (but not optimal) latency for interactive data browsing, queries over small data sets or test queries.
- Hive is not designed for online transaction processing and does not offer real-time queries and row level updates.
- It is best used for batch jobs over large sets of immutable data (like web logs).

Built-In Functions

- round(double a)
- floor(double a)
- ceil(double a)
- rand(), rand(int seed)
- concat()
- substr
- upper
- lower

Built-In Functions - cont.....

- lcase
- trim
- to_date
- year
- month

Aggregate Functions

- count(*)
- sum
- avg
- max
- min

HBase

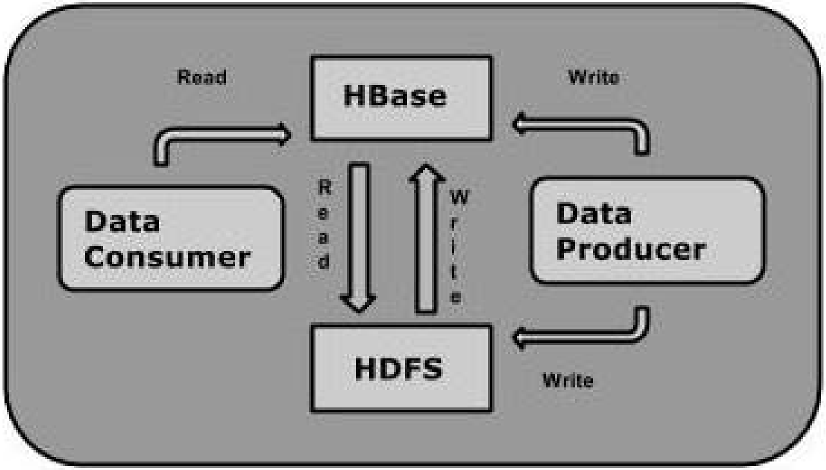
- HBase is a data model that is similar to Google's big table designed to provide quick random access to huge amounts of structured data

Limitations of Hadoop

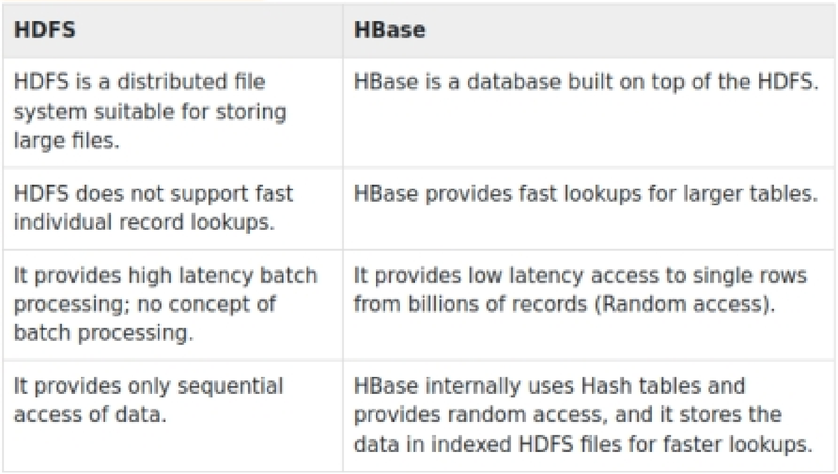
- Hadoop can perform only batch processing, and data will be accessed only in a sequential manner. That means one has to search the entire dataset even for the simplest of jobs.
- A huge dataset when processed results in another huge data set, which should also be processed sequentially. At this point, 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.
- It is a part of the Hadoop ecosystem that provides random real-time read/write access to data in the Hadoop File System.
- One can store the data in HDFS either directly or through HBase. Data consumer reads/accesses the data in HDFS randomly using HBase. HBase sits on top of the Hadoop File System and provides read and write access.


Read
HBase
Write
Data Consumer
Read
Write
Data Producer
HDFS
Write

HBase and HDFS



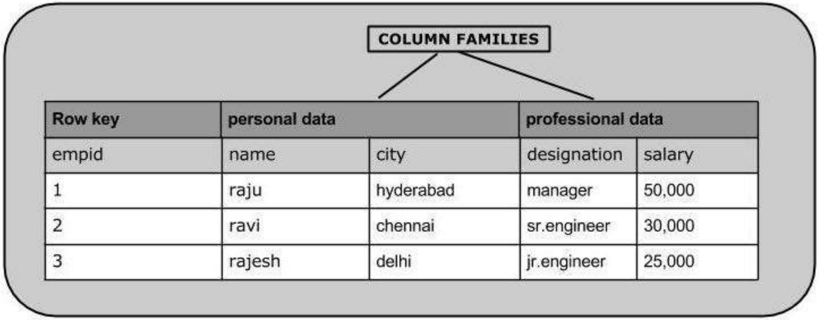
HDFS	HBase
HDFS is a distributed file system suitable for storing large files.	HBase is a database built on top of the HDFS.
HDFS does not support fast individual record lookups.	HBase provides fast lookups for larger tables.
It provides high latency batch processing; no concept of batch processing.	It provides low latency access to single rows from billions of records (Random access).
It provides only sequential access of data.	HBase internally uses Hash tables and provides random access, and it stores the data in indexed HDFS files for faster lookups.

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. Each cell value of the table has a timestamp

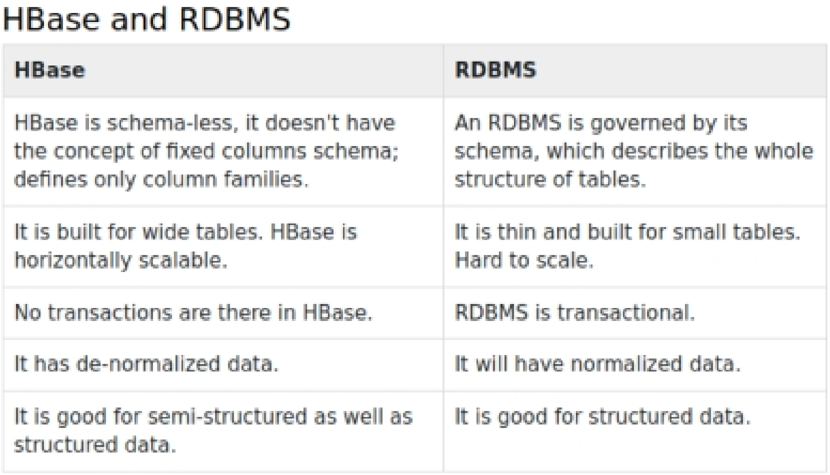
- Table is a collection of rows.
- Row is a collection of column families.
- Column family is a collection of columns.
- Column is a collection of key value pairs.

column-oriented database


COLUMN FAMILIES

Row key	personal data		professional data	
empid	name	city	designation	salary
1	raju	hyderabad	manager	50,000
2	ravi	chennai	sr.engineer	30,000
3	rajesh	delhi	jr.engineer	25,000

HBase and RDBMS



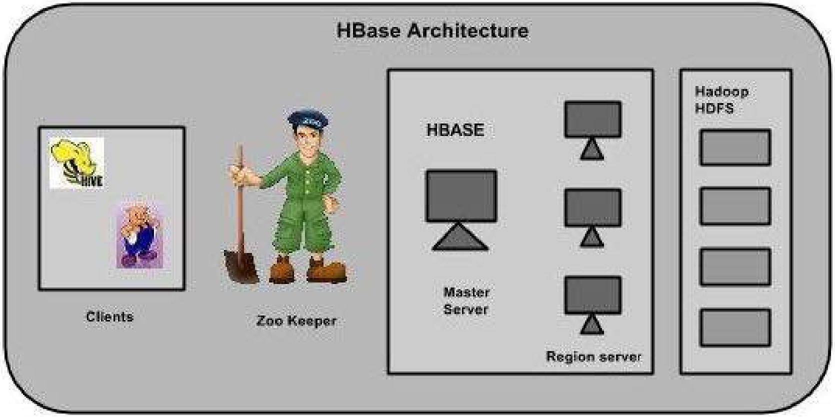
HBase and RDBMS

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.

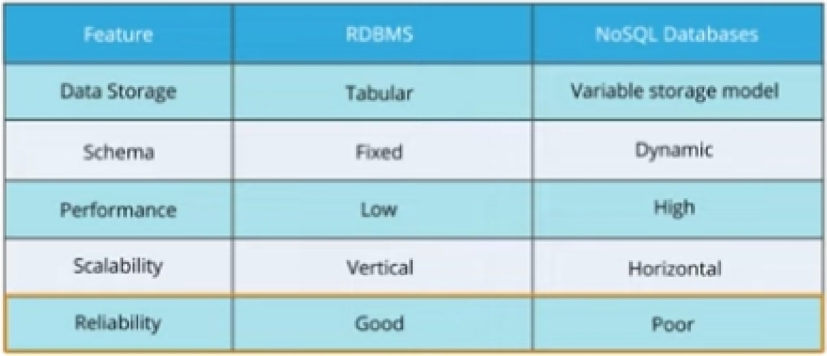
Features of HBase

- HBase is linearly scalable.
- It has automatic failure support.
- It provides consistent read and writes.
- It integrates with Hadoop, both as a source and a destination.
- It has easy java API for client.
- It provides data replication across clusters.

HBase Architecture



RDBMS- NoSql



Feature	RDBMS	NoSQL Databases
Data Storage	Tabular	Variable storage model
Schema	Fixed	Dynamic
Performance	Low	High
Scalability	Vertical	Horizontal
Reliability	Good	Poor

HBase Architecture

HBase has three major components:

- client library
- master server,
- region servers

HBase Shell

- HBase uses the Hadoop File System to store its data.
- It will have a master server and region servers.
- The data storage will be in the form of regions (tables).
- These regions will be split up and stored in region servers.
- The master server manages these region servers and all these tasks take place on HDFS.

General Commands

- status - Provides the status of HBase, for example, the number of servers.
- version - Provides the version of HBase being used.
- table_help - Provides help for table-reference commands.
- whoami - Provides information about the user.

Data Definition Language

- create - Creates a table.
- list - Lists all the tables in HBase.
- disable - Disables a table.
- is_disabled - Verifies whether a table is disabled.
- enable - Enables a table.
- is_enabled - Verifies whether a table is enabled.
- describe - Provides the description of a table.
- alter - Alters a table.
- exists - Verifies whether a table exists.
- drop - Drops a table from HBase.
- drop_all - Drops the tables matching the 'regex' given in the command.
- Java Admin API

Data Manipulation Language

- put - Puts a cell value at a specified column in a specified row in a particular table.
- get - Fetches the contents of row or a cell.
- delete - Deletes a cell value in a table.
- deleteall - Deletes all the cells in a given row.
- scan - Scans and returns the table data.
- count - Counts and returns the number of rows in a table.
- truncate - Disables, drops, and recreates a specified table.
- Java client API -