**PIG:**

Pig is an ETL library for Hadoop, and it generates MapReduce jobs just like Hive does.pig is na abstraction.

When to use Pig:

* When you have processes that are ETL-like
* data cleansing , To clean data. It's really common in big data scenarios that you're bringing in behavioral data, and it might have junk or data that is incorrect, improper or that you don't care about.

using the Pig language against HDFS and Hadoop Platform really makes a lot of sense when you've got a tremendous volume of data, and maybe you just wanna aggregate it up. An example might be, you might get location data from all of your customers worldwide as they go through your stores or go through a certain section of the offering that you have. And maybe you only want the data from a certain time period, from Time A to Time B, and the rest of the data you don't need. So it's a great use of resources to dump all that data into Hadoop cluster, and then use Pig to filter and process that data.

Pig Concepts:

**Data** , Field,Tuple,Bag and Relaation

**Filters**, Filter using logical and relational operators

**Functions** , very rich function library like A) General: AVG,MAX,TOKENIZE B) Relational: FILTER , MAPREDUCE(you can call MR job from pig script and that is very powerful feature).C) String D)Math: ABS,LOG,ROUND E)Loading/Storage: JSONLOADER

**UDFs**, can be written in Java or python to optimize function 1)write the function 2)register the function 3) test the function

**HIVE**

Hive runs on your workstation and converts your SQL query into a series of jobs for execution on the Hadoop cluster. Hive organize data into tables , which provide a means for attaching structure to data stored in HDFS. Metadata – as tables schemas- is stored in database called Measure.

**CREATE TABLE RECORDS (YEAR string,TEMP int, QUALITY int)**

**ROW FORMAT DELIMITED**

**FIELDS TERMINATED BY ‘\T’;**

Here , ROW FORMAT saying each row in the data field is tab-delimited text and rows by new line.

Bucketing in Hive:

LOAD DATA LOCAL INPATH ‘INPUT/NCDC/MICRO-TAB/SAMPLE.TXT’

OVERWRITE INTO TABLE records.

OVERWRITE tells Hive to delete any existing file in the directory for the table, if it is omitted, the files are simply added to the table’s directory(unless they have the same name , in which case they replace the old files).

Running this command tells Hive to put specified local file in its warehouse directory.

Hive tables are stored as directories under Hive’s warehouse directory. Which is controlled by hive.metastore.warehouse.dir property and default to /user/hive/warehouse.thus, the file for the record table is found in /user/hive/warehouse/records directory on local file system.

% ls /user/hive/warehouse/records

Sample.txt

In this case, there is only one file , sample.txt, but in general there can be more and Hive will read all of them when querying the table.

**Execution engine:** MapReduce is default EE but Apache Tez and Spark also possible. Tez and Spark are directed acyclic graph(DAG) engines that offers more flexibility and better performance than MR, in MR intermediate job output is materialized to HDFS but Tez and Spark avoid replication overhead by writing the intermediate output to local disk or even in memore (at request).

**Hive Services:**

* Cli
* Hiveserver2
* Beeline
* Hwi
* Jar
* Metastore

**Hive Clients:** read hiveserver2 service

* Thrift Client <https://stackoverflow.com/questions/40924632/what-is-hiveserver-and-thrift-server>
* JDBC Driver
* ODBC Driver

The Metastore: is central repository of Hive data. Divided into two pieces , a service and backing store for the data.there are three type of Metastore

* Embedded
* Local
* Remote

Usually Partitioning in Hive offers a way of segregating hive table data into multiple files/directories. But partitioning gives effective results when,

* There are limited number of partitions
* Comparatively equal sized partitions

But this may not possible in all scenarios, like when are partitioning our tables based geographic locations like country, some bigger countries will have large partitions (ex: 4-5 countries itself contributing 70-80% of total data) where as small countries data will create small partitions (remaining all countries in the world may contribute to just 20-30 % of total data). So, In these cases Partitioning will not be ideal.

To overcome the problem of over partitioning, Hive provides Bucketing concept, another technique for decomposing table data sets into more manageable parts.

[**http://hadooptutorial.info/bucketing-in-hive/**](http://hadooptutorial.info/bucketing-in-hive/)

[**https://stackoverflow.com/questions/19128940/what-is-the-difference-between-partitioning-and-bucketing-a-table-in-hive**](https://stackoverflow.com/questions/19128940/what-is-the-difference-between-partitioning-and-bucketing-a-table-in-hive)

[**http://myitlearnings.com/bucketing-in-hive/**](http://myitlearnings.com/bucketing-in-hive/)