What is -

* 1. YARN?

In a cluster architecture, Apache Hadoop YARN sits between HDFS and the processing engines being used to run applications. It combines a central resource manager with containers, application coordinators and node-level agents that monitor processing operations in individual cluster nodes.

==================================================================================

* 1. Hive?

Apache Hive is a data warehouse software project built on top of Apache Hadoop for providing data query and analysis. Hive gives an SQL-like interface to query data stored in various databases and file systems that integrate with Hadoop.

==================================================================================

* 1. Pig?

Pig is a high level scripting language that is used with Apache Hadoop. Pig enables data workers to write complex data transformations without knowing Java. Pig's simple SQL-like scripting language is called Pig Latin, and appeals to developers already familiar with scripting languages and SQL.

==================================================================================

* 1. Sqoop?

Sqoop is a tool designed to transfer data between Hadoop and relational database servers. It is used to import data from relational databases such as MySQL, Oracle to Hadoop HDFS, and export from Hadoop file system to relational databases.

==================================================================================

* 1. Metastore in Hive?

Hive Metastore is nothing but the central repository of Apache Hive metadata. Metastore is the central repository of Apache Hive metadata. It stores metadata for Hive tables (like their schema and location) and partitions in a relational database.

==================================================================================

* 1. grunt in Pig?

After invoking the Grunt shell, you can run your Pig scripts in the shell. In addition to that, there are certain useful shell and utility commands provided by the Grunt shell. The Grunt shell of Apache Pig is mainly used to write Pig Latin scripts. Prior to that, we can invoke any shell commands using sh and fs.

==================================================================================

* 1. MapReduce?

Hadoop MapReduce (Hadoop Map/Reduce) is a software framework for distributed processing of large data sets on compute clusters of commodity hardware. It is a sub-project of the Apache Hadoop project. The framework takes care of scheduling tasks, monitoring them and re-executing any failed tasks.

==================================================================================

* 1. mapper?

A mapper class is inherited from the API and we override the map() method.

Mapper task is the first phase of processing that processes each input record (from Record Reader) and generates an intermediate key-value pair. Hadoop Mapper store intermediate-output on the local disk.

==================================================================================

* 1. reducer?

Reducer is a phase in Hadoop which comes after Mapper phase. The output of the mapper is given as the input for Reducer which processes and produces a new set of output, which will be stored in the HDFS. Reducers can run parallel since they are independent of each other. In Reducer we do aggregation or summation computation analysis.

==================================================================================

* 1. map task?

A MapReduce job usually splits the input data-set into independent chunks which are processed by the map tasks in a completely parallel manner. The framework sorts the outputs of the maps, which are then input to the reduce tasks. Typically both the input and the output of the job are stored in a file-system.

==================================================================================

* 1. reduce task?

The Reduce Task is a program running on a machine that executes the reducer function multiple times, serially. If you want your data to actually be processed in parallel, you have to (manually) launch multiple reduce tasks, Hadoop will then divide the work between them.

==================================================================================

* 1. shuffle and sort in MapReduce?

Map reduce makes the guarantee that the input to the reducer is sorted by key. The process by which the system performs the sort, and transfer the map outputs to the reducers as input is known as the shuffle.

==================================================================================

* 1. Impala?

Impala is a MPP (Massive Parallel Processing) SQL query engine for processing huge volumes of data that is stored in Hadoop cluster. It is an open source software which is written in C++ and Java. It provides high performance and low latency compared to other SQL engines for Hadoop.

==================================================================================

* 1. Hadoop client?

Apache Hadoop is an open source framework that supports data-intensive distributed applications. Hadoop has many parts, but two are fundamental:

* MapReduce is the framework that understands and assigns work to the nodes in a cluster. MapReduce divides the application into many fragments of work, each of which can be executed or re-executed on any node in the cluster.
* HDFS (Hadoop File System) spans all the nodes in a Hadoop cluster for data storage

==================================================================================

* 1. FsShell?

FsShell is an interface for users to connect with HDFS.

==================================================================================

* 1. Application Master?

The ApplicationMaster is an instance of a framework-specific library that negotiates resources from the ResourceManager and works with the NodeManager to execute and monitor the granted resources (bundled as containers) for a given application. An application can be a process or set of processes, a service, or a description of work.

==================================================================================

* 1. Container?

A container in YARN is where a unit of work happens in the form of task. A job/application is split in tasks and each task gets executed in one container having a specific amount of allocated resources.

==================================================================================

* 1. YarnChild?

Hadoop YARN launches instances of YarnChild in child VM to execute the actual tasks. Those tasks communicate with their Application Master (AM) through the umbilical interface. All child task related to that particular application master would be on halt state. Hadoop admin should either restart the application master or kill it. NodeManager doesn't terminate the failed Application Master.

==================================================================================

* 1. ApplicationsManager?

The ApplicationsManager is responsible for maintaining a collection of submitted applications. After application submission, it first validates the application’s specifications and rejects any application that requests unsatisfiable resources for its ApplicationMaster (i.e., there is no node in the cluster that has enough resources to run the ApplicationMaster itself). It then ensures that no other application was already submitted with the same application ID—a scenario that can be caused by an erroneous or a malicious client. Finally, it forwards the admitted application to the scheduler.

==================================================================================

* 1. Scheduler?

The core idea behind the scheduler was to assign resources to jobs such that on average over time, each job gets an equal share of the available resources. The result is that jobs that require less time are able to access the CPU and finish intermixed with the execution of jobs that require more time to execute. This behaviour allows for some interactivity among Hadoop jobs and permits greater responsiveness of the Hadoop cluster to the variety of job types submitted.

==================================================================================

* 1. Schema on Read in hive?

A traditional relational database stores the data with schema in mind. It knows that the second column is an integer, it knows that it has 40 columns, etc. Therefore, you need to specify your schema ahead of time and have it well planned out. This is "schema on write" -- that is, the schema is applied when the data is being written to the data store.

Hive (in some cases), Hadoop, and many other NoSQL systems in general are about "schema on read" -- the schema is applied as the data is being read off of the data store.

With schema on read, you just load your data into the data store and think about how to parse and interpret later. At the core of this explanation, schema on read means write your data first, figure out what it is later. Schema on write means figure out what your data is first, then write it after.

==================================================================================

* 1. left outer join in hive?

On defining HiveQL Left Outer Join, even if there are no matches in the right table it returns all the rows from the left table. To be more specific, even if the ON clause matches 0 (zero) records in the right table, then also this Hive JOIN still returns a row in the result. Although, it returns with NULL in each column from the right table.In addition, it returns all the values from the left table. Also, the matched values from the right table, or NULL in case of no matching JOIN predicate.

==================================================================================

* 1. right outer join in hive?

The HiveQL RIGHT OUTER JOIN returns all the rows from the right table, even if there are no matches in the left table. A RIGHT JOIN returns all the values from the right table, plus the matched values from the left table, or NULL in case of no matching join predicate.

==================================================================================

* 1. collections datatype in hive?

Hive and other databases, have more primitive data types like:

* Numeric Types
* TINYINT : 1-byte signed integer, from -128 to 127
* SMALLINT : 2-byte signed integer, from -32,768 to 32,767
* INT : 4-byte signed integer, from -2,147,483,648 to 2,147,483,647
* BIGINT : 8-byte signed integer, from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
* FLOAT : 4-byte single precision floating point number
* DOUBLE : 8-byte double precision floating point number
* DECIMAL
* String Types
* STRING
* VARCHAR
* CHAR
* Date/Time Types
* TIMESTAMP
* DATE
* Misc Types
* BOOLEAN
* BINARY

==============================================================

* 1. scalar datatype in Pig?
* The scalar datatypes that Pig supports are as follows:int : It is signed 32 bit integer.
* long : It is a 64 bit signed integer.
* float : It is a 32 bit floating point.
* double : It is a 63 bit floating pint.
* chararray : It is character array in unicode UTF-8 format.
* bytearray : Used to represent bytes. It is the default data type.
* boolean : to represent true/false values

==================================================================================

* 1. WebUI for NameNode

Web UI to look at current status of HDFS, explore file system. The default address of namenode web UI is <http://localhost:50070/>.

==================================================================================

* 1. WebUI for ResourceManager

The  address of namenode web UI is <http://localhost:8088/>. You can open this address in your browser and check the ResourceManager information.

==================================================================================

* 1. WebUI for JobHistoryServer

The  address of namenode web UI is <http://localhost:19888/jobhistory.You> can open this address in your browser and check the JobHistoryServer information.

==================================================================================