1. Give a brief description of the Hadoop eco system?

Apache Hadoop is a framework that allows for the distributed processing of large data sets across cluster of commodity hardware/computers using a simple programming model of Map & Reduce

Hadoop ecosystem comprises of services like HDFS, Map reduce for storing and processing large amount of data sets. In addition to services there are several tools provided in ecosystem to perform different type data modeling operations. Ecosystem consists of hive for querying and fetching the data that's stored in HDFS.

Similarly ecosystem consists of Pig for data flowing language and to implement some map reduce jobs. For data migration and job scheduling, we use some more tools in Hadoop ecosystem.

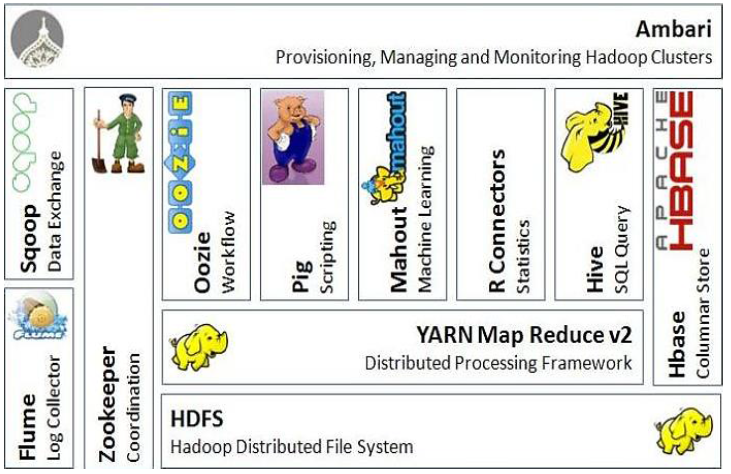
In order to handle large data sets, Hadoop has a distributed framework which can scale out to thousands of nodes. Hadoop adopts Parallel Distributed Approach to process huge amount of data. The two main components of Apache Hadoop are HDFS (Hadoop Distributed File System) and Map Reduce (MR). The basic principle of Hadoop is to write once and read many times.

Hadoop Core Components:

* Hadoop Common
* HDFS
* YARN
* MapReduce

The sub components are:

HBase, Hive, Pig, Sqoop, Flume, Oozie, Zookeeper, Avro, Tez, HCatalog,



2. What is meant by distributed computing and what are the different types of scaling available?

Distributed computing is a computing concept that, in its most general sense, refers to multiple computer systems working on a single problem. In distributed computing, a single problem is divided into many parts, and each part is solved by different computers. As long as the computers are networked, they can communicate with each other to solve the problem. If done properly, the computers perform like a single entity.

The ultimate goal of distributed computing is to maximize performance by connecting users and IT resources in a cost-effective, transparent and reliable manner. It also ensures fault tolerance and enables resource accessibility in the event that one of the components fails.

Scaling is growing an infrastructure (compute, storage, and networking) larger so that the applications riding on that infrastructure can serve more people at a time. When architects talk about the ability of an infrastructure design to scale, we mean that the design as conceived is able to grow larger over time without a wholesale replacement. The terms “scale up” and “scale out” refer to the way in which the infrastructure is grown.

Scale Up: Adding resources to a single node in a system

Scaling up is taking what you’ve got, and replacing it with something more powerful. From a networking perspective, this could be taking a 1GbE switch, and replacing it with a 10GbE switch. Same number of switchports, but the bandwidth has been scaled up via bigger pipes. The 1GbE bottleneck has been relieved by the 10GbE replacement.

Scaling up is a viable scaling solution until it is impossible to scale up individual components any larger. For example, 10GbE is a practical limit for uplinking hosts to the network until such time as 25GbE and higher ports are readily available on hosts. In that context, what happens when 10GbE is no longer enough bandwidth for the uplinked host? Rather than scaling up, you scale out.

Scale out: Adding more nodes to a system

Scaling out takes the infrastructure you’ve got, and replicates it to work in parallel. This has the effect of increasing infrastructure capacity roughly linearly.

3. What is meant by commodity hardware in real time and what are its advantages with reference to enterprise?

commodity hardware involves low-cost desktop computers or workstations that are IBM-compatible and can run operating systems like Microsoft Windows, Linux and DOS without additional software or adaptations. These hardware pieces can be connected and integrated to form more sophisticated computing environments without the purchase of a lot of additional high-design hardware.

4. What is meant by real time and how data is collected in the real time and what are the options for analyzing streaming data?

Real-time data is information that is delivered immediately after collection. There is no delay in the timeliness of the information provided. Real-time data is often used for navigation or tracking.

Data can be collected in real time using Streaming data, which continuously arrives into the system and needs to be processed before the next interval or its expiry.

Apache Spark and Storm are some of the tools used to process streaming data

5. What is the difference between HDFS blocks and input splits?

A block is a hard division of data at the block size. So if the block size in the cluster is 128 MB, each block for the dataset will be 128 MB except for the last block which could be less than the block size if the file size is not entirely divisible by the block size. So a block is a hard cut at the block size and blocks can end even before a logical record ends.

Blocks are physical chunks of data store in disks where as InputSplit is not physical chunks of data.  It is a Java class with pointers to start and end locations in blocks. So when Mapper tries to read the data it clearly knows where to start reading and where to stop reading. The start location of an InputSplit can start in a block and end in another block.