**Difference between Hadoop1 and Hadoop2 ?**

* As Hadoop 1 introduced prior to Hadoop 2 so has some less components and APIs as compare to that of Hadoop 2.

On other hand Hadoop 2 introduced after Hadoop 1 so has more components and APIs as compare to Hadoop 1 such as YARN API,YARN FRAMEWORK, and enhanced Resource Manager.

* Hadoop 1 only supports MapReduce processing model in its architecture and it does not support non MapReduce tools.

On other hand Hadoop 2 allows to work in MapReducer model as well as other distributed computing models like Spark, Hama, Giraph, Message Passing Interface) MPI & HBase coprocessors.

* Map reducer in Hadoop 1 is responsible for processing and cluster-resource management.

On other hand in case of Hadoop 2 for cluster resource management YARN is used while processing management is done using different processing models.

* As Hadoop 1 is prior to Hadoop 2 so comparatively less scalable than Hadoop 2 and in context of scaling of nodes it is limited to 4000 nodes per cluster

On other hand Hadoop 2 has better scalability than Hadoop 1 and is scalable up to 10000 nodes per cluster.

* Hadoop 1 is implemented as it follows the concepts of slots which can be used to run a Map task or a Reduce task only.

On other hand Hadoop 2 follows concepts of containers that can be used to run generic tasks.

* Initially in Hadoop 1 there is no support for Microsoft Windows provided by Apache.

On other hand with an advancement in version of Hadoop Apache provided support for Microsoft windows in Hadoop 2.

**In Hadoop2 why the block size is 128 MB ?**

* To reduce the disk seeks (IO). Larger the block size, lesser the file blocks. Thus, less number of disk seeks. And block can transfer within respectable limits and that to parallelly. HDFS have huge data sets, i.e. terabytes and petabytes of data. If we take 4 KB block size for HDFS, just like Linux file system, which have 4 KB block size. Then we would be having too many blocks and therefore too much of metadata. Managing this huge number of blocks and metadata will create huge overhead. Which is something which we don’t want. So, the block size is set to 128 MB

**Why the namenode is relay on memory rather than datanode ?**

* Name Node only store metadata which is related to the different blocks and because of this reason it needs high memory space. Data Nodes don’t need large memory space.

**Suppose you have 10 PB of data. Metadata is actually store object of file and folder ----> each obj 200 B.**

**How much min Namenode RAM memory you need for your data node in a cluster to manage the metadata?**

**Estimate minimum Namenode RAM size for HDFS with 10 PB capacity, block size 64 MB, average metadata size for each**

**block is 200 B, replication factor is 3.**

* 10 PB/(64MB \*3) \*200B = (10 \* 10^15)/(64\* 10^6 \* 3)\*200 B = 10^10/(64\*3)\* 300B = 1.5625e10 B

**At the time of failure, Which will recover first DataNode or NameNode ?**

* In my opinion, NameNode.