Assignment-20.3:

Explain in brief Writable and Writable Comparable in Hadoop with an example.

a.) A serializable object which implements a simple, efficient, serialization protocol, based on DataInput and DataOutput.

b.) Any key or value type in the Hadoop Map-Reduce framework implements this interface.

c.) Writable is an interface which is used as wrapper for all the primitive datatypes. So that the transfer of the files over the network will become very difficult.

d.) This ease of transferring the files over the network is done with the help of the serialization of the data.

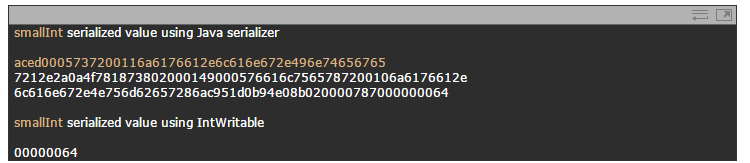
e.) The writable datatypes we first introduced in Hadoop to address the above problem. This is why we are using the Intwritable, Text, LongWritable etc. as the datatypes.

Definition of Serialization:

Serialization is nothing but converting the raw data into a stream of bytes which can travel along different networks and can reside in different systems

Importance of writable and its edge over serialization object of java

Serialization is important in Hadoop because it enables easy transfer of data. If Writable is not present in Hadoop, then it uses the serialization of Java which increases the data over-head in the network as shown in fig.



Example of Custom Writable Type:



There is no compareTo method to sort the key value pairs in the frame work. This Drawback can be overcome in the Writable Comparable.

**Writable Comparable:**

**WritableComparable**interface is just a subinterface of the Writable and java.lang.Comparable interfaces. For implementing a WritableComparable we must have compareTo method apart from readFields and write methods, as shown below

There are three methods in the Writable Comparable.

a.) void readFields (DataInput in);

b.) void write (DataOutput out);

c.) int compareTo (WritableComparable o)

public interface WritableComparable extends Writable, Comparable

{

void readFields(DataInput in);

void write(DataOutput out);

int compareTo(WritableComparable o)

}

Custom writable we created needs to be compared with other datatypes then we have to use the writable comparable. Mostly used when are creating it for keys.

Suppose we are using writable for a key the shuffling of the keys will not happen in shuffle and sort

Mostly shuffling of the keys is to be done. Hence whenever we are dealing with the key that time we are referring writable comparable.