Hadoop Data Types:::

Why Hadoop Datatypes is needed ?? Why Can’t we use Already JAVA Existed Datatypes.

\*\* To Understand this first Try To Understand About the Serialization .

**What Is Serialization ??**

**\***\* When two process intercommunicates (for Ex: Map Communicates with Reduce Phase ) then in that Case , Data is being transferred in terms of Objetcs.

\*\* Serialization is the process of turning structured objects into Byte Stream for the transmission over the network or for writing to the persistent storage which eventually would be read by Another Process.

\*\* On the other hand Deserialization is the process in which the Received byte Streams are processed to read it…

It is the process of turning Byte Stream back into a series of Structured Objects.

In this Process, Communication happens by RPC (Remote Procedure Calls ) in Hadoop.

**Features of Serialization::**

Features that are needed to make RPC effective are::

1. **Compact :** Messages sent over Network should be as small as possible.

The smaller the Data transfer – better would be the efficiency.”

1. **Fast::** Serialization and Deserilization process should happen Quickly.

( If Data would be small then Serializations and Deserialization automatically will be fast.

1. **Extensible::**Process changes over the Time Period and it should be able to meet the newer requirements..
2. **Interoperable::** It is the desired that process written in One Language can Communicate with Process written in Other Language.

Ex: MAP function is written in JAVA and Reduce would be written into other Languages Like Scala, Python …

So, In this scenario as well the serialization framework should be effective…

Now, We understand How HADOOP Works under RPC and Serization is an Important Feature for its Efficiency…

**BUT WHY WE NEED NEW DATA TYPES ??? Can’t We use already Java Serialzable Framework**

Answer To this Question is JAVA has Inbuilt Serialization method which has some sort coming….

First and Most Important:: It was’t Compact. It has overheads when data was Serialized.

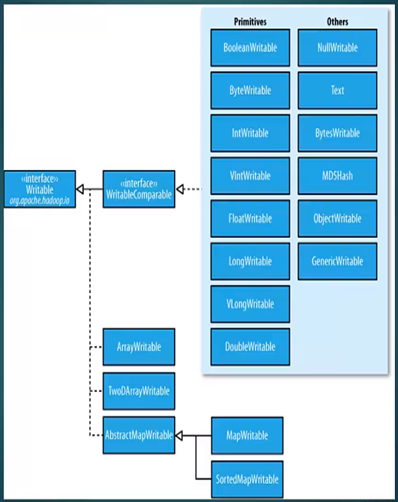
Java Serialization sends the meta Data (like the class Definition) along with the data Set. This was considerably increase the size of Serialized data (the Data Sent)..

As well as processing Time is also Increased… It was Basically Designed for general purpose IPC mechanism.

* It has lots of overheads
* Splitting and random access was difficult
* General Purpose Mechanism.

Now Hadoop Serialization framework assumes that the client already Knows about the data to expected from the Sender.

This decreases the a lots of Overhead and this writable serialization framework was redesigned….





Even Custom Writable class can be done by Extending the writable comparable Interface But in this case the following functions should be invoked::

Write ()

ReadFiled()

CompareTo()

HashCode()

Equals()

ToString()

Because they are going to be inherited from the Interface or are being used in sort or Shuffle stage..