Questions:

1. How the data flow happens through disk and memory during the computation

Ans: The data that is processed by MapReduce is basically stored in Hadoop Distributed File System (HDFS). As the name implies, HDFS divides the data into several blocks which are then stored in a distributed fashion.

The various steps involved in a typical data flow in MapReduce are as follows:

i. In the **first step**, one of the main aims to be achieved is parallelism. So, each block is processed by one mapper. This mapper runs on all the nodes in the cluster and helps achieve parallelism.

ii. As a part of the **second step**, the output that is generated by the mapper in the first step is stored onto the local disk. This output that is stored is only temporary, and not stored onto HDFS since it will result in the generation of multiple copies which will in turn require maintenance and might lead to redundancy.

iii. In the **third step**, the output of the mapper is shuffled to the reducer. Actual physical movement of the data is involved in this step. Consequently, this step takes place over the network.

iv. Once the execution of the mappers is finished and their output is shuffled onto the reducers, the intermediate output is sorted and merged as a part of the **fourth step**. This is then provided to the reducer as the input.

v. As noted above, the input to the reducer (which is the last phase) is from the mappers. The output generated from the reducer is the final output; and hence noted down onto HDFS. This concludes the **fifth** and final step of the data flow.

Reference:

http://data-flair.training/blogs/hadoop-mapreduce-flow-how-data-flows-in-mapreduce/#a_MapReduce_Data_Flow

2. Transformation of data during the computations, i.e., data type of key, value

Ans: The input to the mapper is the <key, value> pairs. Here we take the number, for e.g., 0.91 from the input file as the key as well as the value. Thus, the data type of the key is text and that of the value is double.

3. The data structure used to transfer between Map and Reduce phases

Ans: In order to handle various things the *Hadoop way,* it uses various other counterparts to what is normally used in traditional Java programming. Thus, *Text* is used instead of *String*. *Comparable* and *Writable* are also used. *Writable* data structure is used for storing onto the local disk and for data transfers. The advantage of using *Writable* is that it can serialize the Hadoop objects in a light manner, unlike the *Serializable* interface in Java which would prove to be too heavy. Also, the *Text* data type is used for the key and *DoubleWritable* for value.

References:

http://stackoverflow.com/questions/19441055/why-does-hadoop-need-classes-like-text-or-intwritable-instead-of-string-or-integ