Assignments-22.3:

Explain in brief

● Sequence File Format

● NLine Input Format

● DB Input Format

● DB Output Format

**Sequence File Format:**

1.) Sequence files are flat files containing key, value pairs.

2.) A very common use case when designing ingestion systems is to use Sequence files as containers and store any file related metadata (filename, path, creation time etc.) as the key and the file contents as the value.

3.) A sequence file has a header which contains information on the key/value class names, version, file format, metadata about the file and sync marker to denote the end of the header.

4.) The header is followed by the records which constitute the key/value pairs and their respective lengths.

5.) A Sequence file can be have three different formats: An Uncompressed format, a Record Compressed format where the value is compressed and a Block Compressed format where entire records are compressed.

There are 3 different SequenceFile formats:

1. Uncompressed key/value records.
2. Record compressed key/value records - only 'values' are compressed here.
3. Block compressed key/value records - both keys and values are collected in 'blocks' separately and compressed.

**NLineInputFormat:**

1.) NLineInputFormat which splits N lines of input as one split.

2.) NLineInputFormat in Hadoop is another form of TextInputFormat where the keys are byte offset of the line and values are contents of the line.

3.) Each mapper receives a variable number of lines of input with TextInputFormat and KeyValueTextInputFormat and the number depends on the size of the split and the length of the lines.

4.) And if we want our mapper to receive a fixed number of lines of input, then we use NLineInputFormat.

5.) N is the number of lines of input that each mapper receives. By default (N=1), each mapper receives exactly one line of input.

6.) If N=2, then each split contains two lines. One mapper will receive the first two Key-Value pairs and another mapper will receive the second two key-value pairs

**DB Input Format:**

1.) The DBInputFormat component provided in Hadoop 0.19 finally allows easy import and export of data between Hadoop and many relational databases, allowing relational data to be more easily incorporated into your data processing pipeline.

2.) DBInputFormat uses JDBC to connect to data sources. Because JDBC is widely implemented, DBInputFormat can work with MySQL, PostgreSQL, and several other database systems.

3.) The DBInputFormat is an InputFormat class that allows you to read data from a database.

4.) An InputFormat is Hadoop’s formalization of a data source; it can mean files formatted in a particular way, data read from a database, etc.

5.) DBInputFormat provides a simple method of scanning entire tables from a database, as well as the means to read from arbitrary SQL queries performed against the database.

**DB Output Format:**

1.) DBOutputFormat accepts <key,value> pairs, where key has a type extending DBWritable. Returned RecordWriter writes only the key to the database with a batch SQL query.

2.) The DBOutputFormat writes to the database by generating a set of INSERT statements in each reducer.

3.) The reducer’s close() method then executes them in a bulk transaction.

4.) Performing a large number of these from several reduce tasks concurrently can swamp a database.

5.) If you want to export a very large volume of data, you may be better off generating the INSERT statements into a text file, and then using a bulk data import tool provided by your database to do the database import.