Comparison

The Goal is to understand what files between them are suitable:

* Get read fast (mainly the case of row based storage)
* Get written fast (mainly the case row based storage)
* Executed quickly a OLAP or OLTP query
* Be splittable i.e. multiple task can run parallel on parts of file
* Support Schema evolution, allowing us to change schema of file
* Support advanced compression through various available compression codecs (Bzip2, LZO, Sappy). That allow to reduce data storage space on disk, increase the performance of lecture and readable on the disc and also the transfer speed in the network.

**# Row based file format**

CSV

It’s commonly used to exchange tabular data between systems using plan text, splittable , less compressible file.

Advantage : human-readable and easy to edit manually; provide a straightforward information schema; processed by almost all existing applications; simple to implement and parse, write performance but slow to read

Drawback : don’t manage Null value and not standard for Big data

JSON (JavaScript object notation) and XML

Json is represented as key-value pairs in partially structured format. It store data to the hierarchical format. It is self-describing and readable. It is smaller in term of size. Json is often used in network communication; also in serialize of deserialize data. His kind of storage (row based data ) can be optimized by containing parquets or avro format.

Many Batches or Stream data processing models natively support JSON serialization and deserialization. It attach metadata to the data in each record.

Advantage : It support hierarchical structures and a lists of objects; widely used for NoSQL

Drawback : Xml is less readable compared to Json

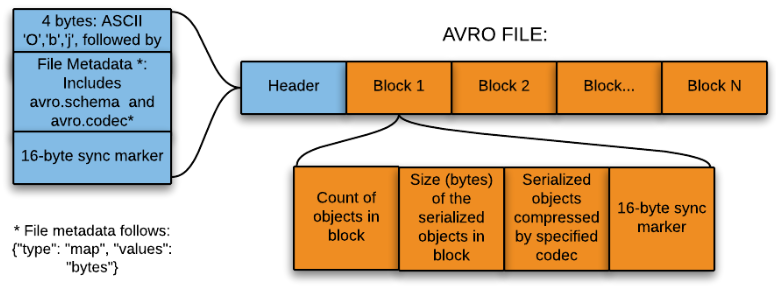
AVRO

It is a row-based format that is highly splittable and compressible. It also described as a data serialization system and deserialization. The schema is stored in JSON format while the data is stored in *binary format*, minimizing file size and maximizing efficiency and compact. This following propriety allow it to can be supported in many different programming languages ((python, C, C++, …) .

It’s a good candidate for data storage in Hadoop ecosystem. Has Enough capacity to manage the schema evolution (at different time and independently), can be used in streaming data.

Advantage : fast data serialization and deserialization which can deliver very good ingestion performance, has a sync marker to separate the block (splittable), typically used to write-heavy workload because rows can be added simply and quickly. His key feature is the robust support for data schema that changes over time (**schema evolution**) , average read/write performance. It’s the best choice if your data schema change frequently (files can renamed, added, deleted, while old files can still be read with the new schema)

Drawback :



**# Column based file format**

The advantage of these types of storage is based to avoid I/O, good performance of queries that only access certain columns, more efficacy for the compression because usually the data looks the same.

Parquet

Parquet is splittable and is a binary file containing metadata about their content. The column metadata for a Parquet file is stored at the end of the file, which allows for fast, one-pass writing. Parquet is optimized for the write Once read many (WORM). It’s slow to write but very fast to read. It’s compatible with all MapReduce interfaces such as Java, Hive, Pig

Advantage : fastest for reading workflow but very slow to writes, well known in nested data structure complexes, support compression mostly with snappy algorithm

Drawback : immutable, does not support schematic evolution. It cannot be created from streaming data

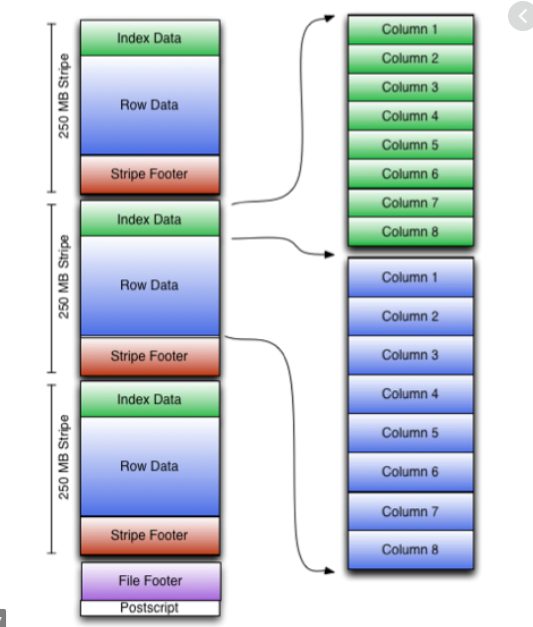
ORC

The ORC format can only support Hive (Suitable for OLAP query), lightweight file and always in compression. It support also the supplementary compression via the use of LZO, Snappy.

It a column storage data with stripe, reduce the size of the original data up to 75%

Advantage : improves performance reading, writing, and processing in Hive; Optimize Cost storage

Drawback : Can’t be load data directly into ORCFILE, increases CPU overhead by increasing the time it takes to decompress the relational data, Specific version



Protocol Buffer

It is a binary encoding format that allows you to specify a schema for your data using a specification language. It used for serialize of structured data.

Advantage:

Drawback: No splittable, No compressible and don’t support MapReduce

**Summary**

CSV and Json are less suitable for stockage or data analysis. They are also very low to be analysed. In term of serialization, Avro and protocol buffer are more appropriate.

Avro is more generally fastest to write while parquet is more easily to read the column.

Json is a standard for communication on the web.

Parquet and Avro are more optimized for Big data cause due to their splittability, compression support .

In practice, row-oriented storage layouts are well-suited for OLTP-like workloads.

Column-oriented storage layouts are well-suited for OLAP-like workloads (e.g., data warehouses) which typically involve a smaller number of highly complex queries over all data (possibly terabytes).