15-HadoopFileFormats

August 11, 2020

1 Hadoop File Formats

Format Wars: From VHS and Beta to Avro and Parquet

1.1 Feather

For light data, it is recommanded to use Feather. It is a fast, interoperable data frame storage that comes with bindings for python and R.

Feather uses also the Apache Arrow columnar memory specification to represent binary data on disk. This makes read and write operations very fast.

[1]:
```
import feather
import pandas as pd
import numpy as np
arr = np.random.randn(10000) # 10% nulls
arr[::10] = np.nan
df = pd.DataFrame({'column_{0}'.format(i): arr for i in range(10)})
feather.write_dataframe(df, 'test.feather')
```

[2]:
```
df = pd.read_feather("test.feather")

df.head()
```

[2]:

	column_0	column_1	column_2	column_3	column_4	column_5	column_6	column_7	column_8	column_9
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	-2.037747	-2.037747	-2.037747	-2.037747	-2.037747	-2.037747	-2.037747	-2.037747	-2.037747	-2.037747
2	-0.609178	-0.609178	-0.609178	-0.609178	-0.609178	-0.609178	-0.609178	-0.609178	-0.609178	-0.609178
3	-0.079906	-0.079906	-0.079906	-0.079906	-0.079906	-0.079906	-0.079906	-0.079906	-0.079906	-0.079906
4	-0.207763	-0.207763	-0.207763	-0.207763	-0.207763	-0.207763	-0.207763	-0.207763	-0.207763	-0.207763

1.2 Parquet file format

Parquet format is a common binary data store, used particularly in the Hadoop/big-data sphere. It provides several advantages relevant to big-data processing:

- columnar storage, only read the data of interest
- efficient binary packing
- choice of compression algorithms and encoding
- split data into files, allowing for parallel processing
- range of logical types
- statistics stored in metadata allow for skipping unneeded chunks
- data partitioning using the directory structure

1.3 Apache Arrow

Arrow is a columnar in-memory analytics layer designed to accelerate big data. It houses a set of canonical in-memory representations of flat and hierarchical data along with multiple language-bindings for structure manipulation.

https://arrow.apache.org/docs/python/parquet.html

The Apache Parquet project provides a standardized open-source columnar storage format for use in data analysis systems. It was created originally for use in Apache Hadoop with systems like Apache Drill, Apache Hive, Apache Impala, and Apache Spark adopting it as a shared standard for high performance data IO.

Apache Arrow is an ideal in-memory transport layer for data that is being read or written with Parquet files. PyArrow includes Python bindings to read and write Parquet files with pandas.

Example:

```
import pyarrow as pa

hdfs = pa.hdfs.connect('svmass2.mass.uhb.fr', 54311, 'navaro_p')
```

[3]:
```
import pyarrow.parquet as pq
import numpy as np
import pandas as pd
import pyarrow as pa
```

[4]:
```
df = pd.DataFrame({'one': [-1, np.nan, 2.5],
                   'two': ['foo', 'bar', 'baz'],
                   'three': [True, False, True]},
                   index=list('abc'))
table = pa.Table.from_pandas(df)
```

[5]:
```
pq.write_table(table, 'example.parquet')
```

[6]:
```
table2 = pq.read_table('example.parquet')
```

[7]:
```
table2.to_pandas()
```

[7]:
```
   one  two  three
a -1.0  foo   True
b  NaN  bar  False
c  2.5  baz   True
```

[8]:
```
pq.read_table('example.parquet', columns=['one', 'three'])
```

[8]:
```
pyarrow.Table
one: double
three: bool
```

[9]:
```
pq.read_pandas('example.parquet', columns=['two']).to_pandas()
```

[9]:
```
   two
a  foo
b  bar
c  baz
```