ECE4721J — Methods and Tools for Big Data

Million Song Dataset (MSD) Analysis Tools

p1team-01

UM-JI (Summer 2022)

July 26, 2022

Overview

- Maven-managed Java Project
  - Cross-platform
  - Easy to install new packages (compared with c++)
  - Easy to manage different packages (compared with Python `import`)
- Avro
  - Easy to be integrate into Java project as part of Apache Ecosystem
  - Can be easy accessd and processed by Drill and Spark easily
  - Compact small files together to avoid waste of memory in HDFS
  - More freedom in data retrieve

HDF5 file related feature

HDF5 file related functions are implemented within `H5_parser` class

- `H5_parser.recursivePrintGroup`: Print all the group information stored in selected h5 file
- `H5_parser.printData`: Print all the data with its paths. The compound data will be print separately.
- `H5_parser.printDataType`: Print the data type of each field in the HDF5 file.

Overview

Basically, we provide three kinds of avro compact method.

- `song`: Compact all the information with respect to its field and the final results will be separated into analysis, metadata and musicbrainz in Drill
- `song_summary`: Compact only the information required for Drill process to provide Drill-friendly avro file
- `artists`: Compact only the information required for constructing graph for advanced analysis feature. You can manually generate for a test but not recommended.

Avro related feature

Avro related functions are implemented within `CompactSmallFiles` class

- `CompactSmallFiles.serialize`: compact all the h5 files within the folder in `song` mode
- `CompactSmallFiles.serializeSummary`: compact all the h5 files within the folder in `song_summary` mode
- `CompactSmallFiles.serializeArtists`: compact all the h5 files within the folder in `artists` mode
- `CompactSmallFiles.serializeArtists_N`: compact defined number of h5 files within the folder in `artists` mode
- `CompactSmallFiles.readDir`: store all the h5 file into the avro process class within the folder and print the number of files store

template

Template as a special class see Page 331:

```cpp
#include <iostream>
using namespace std;
template<class TYPE>
class Complex {
public:
Complex(){ R = I = (TYPE)0; }
Complex(TYPE real, TYPE img) {R=real;I=img;}
void PrintComplex() {cout<<R<<'+'<<I<<"i\n";}
private:
TYPE R, I;
};
```

Basic Usage, insert the TYPE with proper datatype :

```cpp
Complex<float> c1; complex<int> c2;
typedef Complex<double> dcplx; dcplx c3;
```

If you turn to cppreference, TYPE is more often written as T.

Standard Template Library

Reference

[1] 1