Hand-Off Document

Team: OAL Parquet + OCI
Date: June 11, 2020
Github: https://github.com/svaddadhi/Apache-Parquet-OCI

Team:

- Phu Le: phvle@ucsc.edu
- Xiaobin Wu: xwu63@ucsc.edu
- Yibo Guo: yguo25@ucsc.edu
- David Hernandez: dherna99@ucsc.edu
- Vishal Vaddadhi: svaddadh@ucsc.edu

Overview:

Our project is centered around creating an Oracle Cloud service that leverages Apache Parquet file format. This is done with data-analyzing operations like filtering, searching, and file conversion on Parquet files. Users simply have to pass in certain credentials when making calls to the server. As a result, users will be able to process Parquet and CSV files on the Oracle Cloud and produce target Parquet files to the cloud.

Usage:

- Requirements:
  - Intellij
  - Gradle
  - Java
- Build:
  - gradle build
  - Intellij build task
- Run:
  - gradle runJar
  - Intellij [parquet-oci] runJar task
- Pre-Usage:
  - Load properties file:
    - parquet-oci/library-sub-project/src/main/resources/config.properties
  - Load ‘drill-jdbc-all-1.17.0.jar’:
    - library-sub-project/build/libs/
  - Apache Drill server running
    - Set host and clusterId in properties file
- Usage: refer to README.md for call examples

Tasks Done:

- Upload/Download to OCI Object Storage
- Natively convert csv to Parquet
- Drill convert csv to Parquet
- Drill filter columns from Parquet
- Drill conditional and unconditional import from CSV/Parquet
- Drill export to CSV/Parquet
- JMH benchmark for native conversion vs. drill conversion
- Helidon application
- Endpoints for convert, download, upload, filter columns

Ongoing Tasks:

- Create a docker container for the service
- Run the service using GraalVM
- Testing for Helidon application
- Deploy service to OCI
- Endpoints for filtering row (search)

Known Issues:

- Dockerfile: the current Dockerfile builds but fails on run
- Native conversion: the target Parquet file does not set the column titles appropriately
- Filtering Rows: the Helidon resource file for filtering rows has not been tested

Troubleshooting:

- Any gradle tasks done on the terminal can be done by commenting out line 87 and lines 119-121 in library-sub-project/build.gradle as well as lines 37-39 in helidon-sub-project/build.gradle
- Some issues regarding missing object files may be fixed using gradle clean
- When ‘Caused by: java.lang.ClassNotFoundException: io.helidon.microprofile.server.Server’ error occurs, it might be fixed by running gradle build in the terminal first
- There may be errors regarding JDBC when using the filter columns and rows through the calls to the Helidon server