HiveServer2 and HPL/SQL Integration

Design Proposal

About HPL/SQL

Procedural SQL is a typical way to develop ETL processes in traditional data warehouses. It allows using flow of control statements, loops, cursors, dynamic SQL, stored procedures and many other advanced constructs to create comprehensive ETL pipelines.

Since Hive often complements existing data warehouses, the design goal of HPL/SQL was not to invent a new procedural language, but to be compatible with all major procedural dialects such as Oracle PL/SQL, SQL Server Transact-SQL, DB2/MySQL/Teradata ANSI SPL in the single tool.

This allows porting existing ETL code to Hive with little or no changes as well as leveraging existing skills and known language in the new environment.

Another feature of HPL/SQL is that it has built-in capability to convert SQL statements on the fly, so with some limitations the existing applications with native SQL code can be just redirected to Hive that can bring significant advantage to existing BI reports, applications.

See more details about HPL/SQL at hplsql.org

Motivation

Currently HPL/SQL can be used as a standalone tool that limits its usage. Most existing applications use procedural SQL invoking the stored procedures through JDBC calls.

This proposal is intended to integrate HiveServer2 and HPL/SQL, so procedural SQL will be available in Hive through JDBC.

Interaction with HPL/SQL

There are several ways to interact with HPL/SQL

1. Call a stored procedure (CALL proc_name)

2. Execute anonymous block (BEGIN stmts END)

3. Execute a SQL script (Arbitrary script containing procedural SQL statements, DDL and DML statement) - Out of scope as it should be implemented at the client side (it is similar to

SQL*Plus in Oracle, ISQL in SQL Server etc - and now available in ./hplsql CLI supplied with Hive)

Challenges

- HPL uses own parser and due to supporting many procedural dialects (acts like SQL “skins”) it seems unreasonable to merge HPL and Hive parsers. HiveServer needs to distinguish between Hive SQL and HPL statements.

- HPL currently uses JDBC connection to execute statements in Hive, and this code has to be rewritten to use invernal Hive classes.

- Stored procedures and blocks can return multiple result sets that needs to be properly handled by the caller.

- Client tools such as Beeline, Hue etc. should be able to pass compound statements as is to HiveServer without splitting them by semicolon.

- Stored procedure source code should be stored on HiveServer2 local storage or in the Hive metastore.

Implementation

Phase 1

- Read stored procedures source code from HiveServer2 local path.
- Dynamically allocate HPL executer for each call

Phase 2

- Add stored procedure support in metastore
- Cache HPL/SQL executor for reuse

Similar to HiveCommandOperation and SQLOperation, HPLSQLOperation class can be used in ExecuteStatementOperation.newExecuteStatementOperation() to execute procedural SQL code.

HPLSQLOperation will instantiate org.apache.hive.hplsql.Hplsql that will execute the procedural code and use SQLOperation to execute SQL statements in Hive.