Parquet

Version

Parquet is supported by a plugin in Hive 0.10, 0.11, and 0.12 and natively in Hive 0.13 and later.

Introduction

Parquet (http://parquet.io/) is an ecosystem wide columnar format for Hadoop. Read Dremel made simple with Parquet for a good introduction to the format while the Parquet project has an in-depth description of the format including motivations and diagrams. At the time of this writing Parquet supports the follow engines and data description languages:

Engines

- Apache Hive
- Apache Drill
- Cloudera Impala
- Apache Crunch
- Apache Pig
- Cascading
- Apache Spark

Data description

- Apache Avro
- Apache Thrift
- Google Protocol Buffers

The latest information on Parquet engine and data description support, please visit the Parquet-MR projects feature matrix.

Parquet Motivation

We created Parquet to make the advantages of compressed, efficient columnar data representation available to any project in the Hadoop ecosystem.

Parquet is built from the ground up with complex nested data structures in mind, and uses the record shredding and assembly algorithm described in the Dremel paper. We believe this approach is superior to simple flattening of nested name spaces.

Parquet is built to support very efficient compression and encoding schemes. Multiple projects have demonstrated the performance impact of applying the right compression and encoding scheme to the data. Parquet allows compression schemes to be specified on a per-column level, and is future-proofed to allow adding more encodings as they are invented and implemented.

Parquet is built to be used by anyone. The Hadoop ecosystem is rich with data processing frameworks, and we are not interested in playing favorites. We believe that an efficient, well-implemented columnar storage substrate should be useful to all frameworks without the cost of extensive and difficult to set up dependencies.

Native Parquet Support

Hive 0.10, 0.11, and 0.12

To use Parquet with Hive 0.10-0.12 you must download the Parquet Hive package from the Parquet project. You want the parquet-hive-bundle jar in Maven Central.

Hive 0.13

Native Parquet support was added (HIVE-5783). Please note that not all Parquet data types are supported in this version (see Versions and Limitations below).

HiveQL Syntax

A CREATE TABLE statement can specify the Parquet storage format with syntax that depends on the Hive version.

Hive 0.10 - 0.12

```
CREATE TABLE parquet_test (
 id int,
 str string,
 mp MAP<STRING,STRING>,
 lst ARRAY<STRING>,
 strct STRUCT<A:STRING,B:STRING>)
PARTITIONED BY (part string)
ROW FORMAT SERDE 'parquet.hive.serde.ParquetHiveSerDe'
 STORED AS
 INPUTFORMAT 'parquet.hive.DeprecatedParquetInputFormat'
 OUTPUTFORMAT 'parquet.hive.DeprecatedParquetOutputFormat';
```

Hive 0.13 and later

```
CREATE TABLE parquet_test (
 id int,
 str string,
 mp MAP<STRING,STRING>,
 lst ARRAY<STRING>,
 strct STRUCT<A:STRING,B:STRING>)
PARTITIONED BY (part string)
STORED AS PARQUET;
```

Versions and Limitations

Hive 0.13.0

Support was added for Create Table AS SELECT (CTAS -- HIVE-6375).

Hive 0.14.0

Support was added for timestamp (HIVE-6394), decimal (HIVE-6367), and char and varchar (HIVE-7735) data types. Support was also added for column rename with use of the flag `parquet.column.index.access` (HIVE-6938). Parquet column names were previously case sensitive (query had to use column case that matches exactly what was in the metastore), but became case insensitive (HIVE-7554).

Hive 1.1.0

Support was added for binary data types (HIVE-7073).

Hive 1.2.0

Support for remaining Parquet data types was added (HIVE-6384).

Resources

- Parquet Website
- Format specification
- Feature Matrix
- The striping and assembly algorithms from the Dremel paper
- Dremel paper
- Dremel made simple with Parquet