大数据开源列式存储引擎Parquet和ORC

2015-05-16 董西成 hadoop123

相比传统的行式存储引擎，列式存储引擎具有更高的压缩比，更少的IO操作而备受青睐（注：列式存储不是万能高效的，很多场景下行式存储仍更加高效），尤其是在数据列（column）数很多，但每次操作仅针对若干列的情景，列式存储引擎的性价比更高。

在互联网大数据应用场景下，大部分情况下，数据量很大且数据字段数目很多，但每次查询数据只针对其中的少数几行，这时候列式存储是极佳的选择，目前在开源实现中，最有名的列式存储引擎是Parquet和ORC，在最近一年内，它们都晋升为Apache顶级项目，可见它们的重要性。本文尝试比较这两种存储引擎。

Apache Parquet

源自于google Dremel系统（可下载论文参阅），Parquet相当于Google Dremel中的数据存储引擎，而Apache顶级开源项目Drill正是Dremel的开源实现。

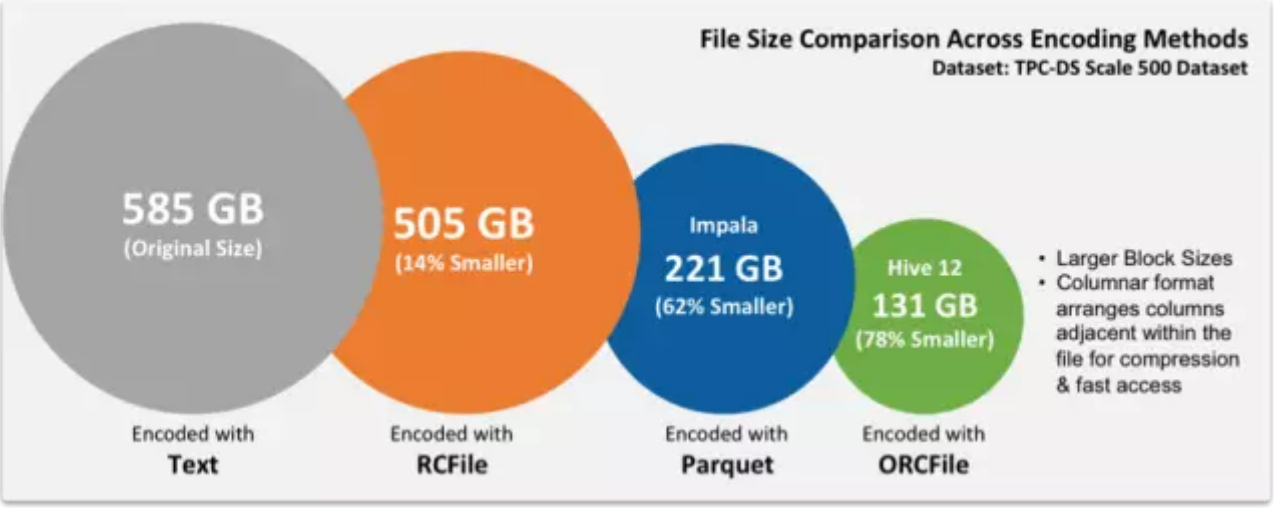
Apache Parquet 最初的设计动机是存储嵌套式数据，比如Protocolbuffer，thrift，json等，将这类数据存储成列式格式，以方便对其高效压缩和编码，且使用更少的IO操作取出需要的数据，这也是Parquet相比于ORC的优势，它能够透明地将Protobuf和thrift类型的数据进行列式存储，在Protobuf和thrift被广泛使用的今天，与parquet进行集成，是一件非容易和自然的事情。 除了上述优势外，相比于ORC，Parquet没有太多其他可圈可点的地方，比如它不支持update操作（数据写成后不可修改），不支持ACID等。

Apache ORC

ORC（OptimizedRC File）存储源自于RC（RecordColumnar File）这种存储格式，RC是一种列式存储引擎，对schema演化（修改schema需要重新生成数据）支持较差，而ORC是对RC改进，但它仍对schema演化支持较差，主要是在压缩编码，查询性能方面做了优化。RC/ORC最初是在Hive中得到使用，最后发展势头不错，独立成一个单独的项目。Hive 1.x版本对事务和update操作的支持，便是基于ORC实现的（其他存储格式暂不支持）。ORC发展到今天，已经具备一些非常高级的feature，比如支持update操作，支持ACID，支持struct，array复杂类型。你可以使用复杂类型构建一个类似于parquet的嵌套式数据架构，但当层数非常多时，写起来非常麻烦和复杂，而parquet提供的schema表达方式更容易表示出多级嵌套的数据类型。

Parquet与ORC对比

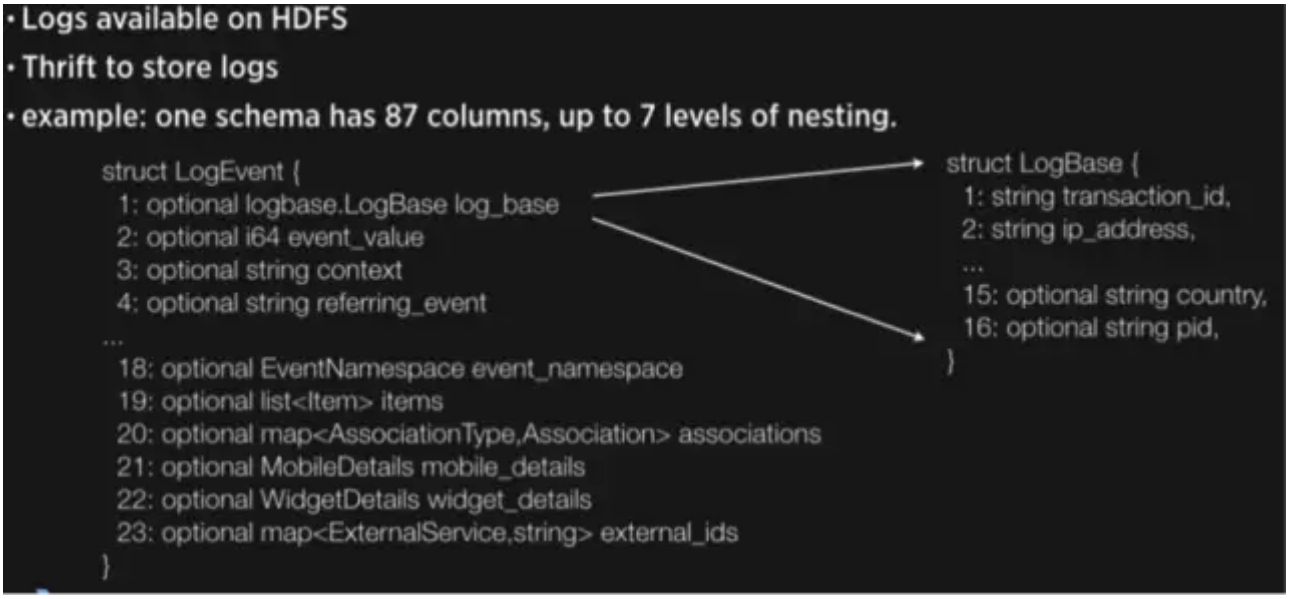
	Parquet (http://orc.apache.org/)	ORC (http://parquet.apache.org/)
现状	Apache 顶级项目，开源，列式存储引擎	
主导公司	Twitter/Cloudera	Hortonworks
开发语言	Java	Java
列编码	支持多种编码，字典，RLE，delta 编码等	支持主流编码，与 parquet 类似
嵌套式结构	支持比较完美	多层级嵌套表达起来复杂，底层未采用 google dremel 类似实现，性能和空间损失较大
ACID	不支持	支持
Update 操作（delete, update 等）	不支持	支持
支持索引（实际上是统计信息）	粗粒度索引，block/group/chunk 级别统计信息	粗粒度索引，file/Stripe/row 级别统计信息，不能精确到列建索引
查询性能	ORC 稍高，可以看 netflix 对比：http://techblog.netflix.com/2014/10/using-presto-in-our-big-data-platform.html	
数据压缩能力	见图	
支持的查询引擎	Apache Drill/impala	Apache hive



总结

目前在互联网领域，列式存储已经逐步被用于各种产品线中，比如twitter已经将部分数据格式转换为parquet，所占空间和查询时间减少了约1/3（来源：https://adtmag.com/articles/2015/04/28/apache-parquet.aspx ）。在

Twitter，日志格式使用thrift描述，使用Parquet存储，下图是一个典型的数据格式描述，共有87个字段，7层嵌套关系。



阅读原文