Apache Drill初探

陈镜融
14307130118@fudan.edu.cn

April 30, 2018

1 Abstract

这篇文章首先介绍了Apache Drill的体系结构和查询执行流程，并介绍了Apache Drill的相关替代软件。之后，文章详细的展示了Drill安装配置和运行的流程，并用1GB的TPC-H数据集在5个节点的集群上进行基准测试。最后用一些简单的测量方法，对查询的瓶颈进行了分析，并解释了对于TPC-H数据集，分布式相比于单机性能提升的关键。

2 Introduction

Apache Drill[3]是一个分布式的低延时大数据查询引擎，它能够查询结构化和半结构化的数据。Apache Drill受到了Google的Dremel[5]启发。如同GFS、BigTable、MapReduce分别对应HDFS、HBASE和Hadoop MapReduce，Drill可以看作是Dremel的开源实现。

设想我们手头具有大量的电子商务数据，而我有一些想法想要验证，因此需要查询和分析这些数据。利用传统的大数据分析方法，我们可以使用MapReduce，或者是Hive。然而利用MapReduce开发成本较高，不适合在线分析，Hive的查询语句HiveQL支持的语法具有局限性。另外，MapReduce和Hive都具有普通查询等待时间太高的诟病。因此Dremel出现了，它提供了可扩展的，大数据交互式分析功能。使用Dremel，可以将常规查询的处理时间从分钟级缩减到秒级。

Dremel作为Google大数据基础设施之一，只提供给公司内部使用。我们能找到一些替代品。Google对外提供了产品Big Query[1]，这款产品可以在Google Cloud Platform上以SaaS的形式被用户调用，因此我们还是无法得到其源代码，以在自己的平台部署。类似的Amazon提供了Amazon Redshift[2]，它是一款数据仓库解决方案，但一样没有开源。Facebook提供了分布式大数据SQL查询引擎Presto[6]，这款软件与Dremel和Apache Drill有着类似的体系结构设计，但其版本才发行到0.1版本。另外，同样是开源软件，还有一款与Drill非常类似的Apache Impala[1]，它主要用于在Hadoop上运行SQL，关于两者的对比我们可以参考`https://db-engines.com/en/system/Apache+Drill%3BHive%3BImpala`。最终我们选取在设计上参考了Dremel的Apache Drill来进行实验和研究。

[1] `https://cloud.google.com/bigquery/`

3 Design

3.1 Core Modules

Drill包括了一个叫做Drillbit的守护进程。Drillbit分布式得运行在集群的任意节点上，用来接收客户端的查询请求，并将请求处理，分发执行计划，并最终将结果收集起来返回给客户端。

Drill使用ZooKeeper来作为协商服务，ZooKeeper中维护了当前所有Drillbit连接信息。当一个Drillbit进程加入时，它可以通过ZooKeeper找到集群中所有其他的Drillbits，并直接和其他Drillbit进行连接。

如图1，一个Drillbit包含如下关键组件

- **PRC end point** 一个RPC end point提供了一个低开销的基于protobuf的PRC协议，来接收查询请求并与其他drillbit通信
- **SQL parser** 它用来解析查询请求，并输出一个语言无关的，逻辑计划来表示这个查询
- **Storage plugin interfaces** 他是对Drill使用的数据源的抽象

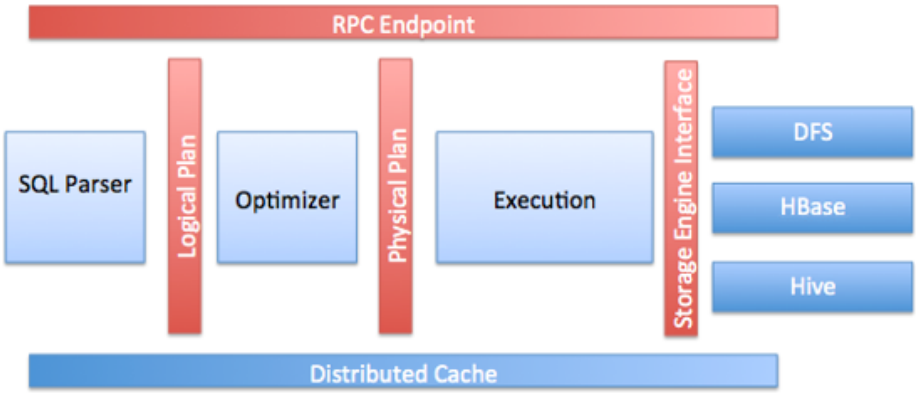

Figure 1: DrillbitModules

3.2 Query Execution

Drill是一个灵活的，可靠的SQL查询引擎。它能够接收多种来源数据和多种来源的请求比如JDBC，ODCBC，REST interface和C++和Java API。

当客户端查询时，收到查询请求的Drillbit进程成为Foreman，它与ZooKeeper通信得到集群里所有在线的Drillbit进程，所有Drillbit进程之间没有主从概念，每一个Drillbit都包含了所有功能。

Foreman收到请求之后，将请求解析为逻辑计划(logical plan)，逻辑计划经过优化器之后，结合查询优化和数据局部性，生成具体的物理计划(physical plan)或者叫做执行计划(execution plan)。物理计划生成后，会被分成具体的执行计划片段(executional plan fragments)，这些分段定义了每个Drillbit执行哪些执行计划片段。如图2

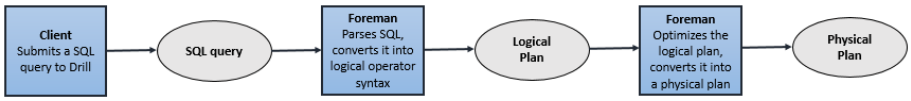

Figure 2: client-phys-plan

3.3 Columnar Storage

Drill和Dremel一样采用列式存储。Drill默认采用Parquet[4]格式进行存储。Parquet将数据按列组织在一起，可以有效降低I/O，有利于数据压缩，并且支持向量运算，从而获得更好的扫描性能。

4 Installation and Configuration

参见`https://github.com/crazyboycjr/apache-drill-tpch-experiments`

5 Query Example

参见`https://github.com/crazyboycjr/apache-drill-tpch-experiments`

6 Experiments

参见`https://github.com/crazyboycjr/apache-drill-tpch-experiments`

7 Summary

所有安装配置和运行的代码全都整理过之后，放在上述GitHub repository里，为了不污染生产环境，所有程序均在Container中运行。README文档阐释了安装配置和运行测试程序的详细过程，保留了全部运行脚本，力求读者能够根据文档复现这个过程。文档中另外还记录了配置过程中遇到的两个坑，其中一个是软件bug，即软件行为和配置行为不一致。最后我尝试用一些简单的测试方法，从计算（CPU），网络和存储（磁盘I/O）角度对查询瓶颈进行了分析。得出了在TPC-H数据集上磁盘I/O是主要瓶颈的结论。

Reference

[1] Bittorf, M., Bobrovytsky, T., Erickson, C., Hecht, M. G. D., Kuff, M. J. I. J. L., Leblang, D. K. A., Robinson, N. L. I. P. H., Rus, D. R. S., Wanderman, J. R. D. T. S., and Yoder, M. M. Impala: A modern, open-source sql engine for hadoop. In *Proceedings of the 7th Biennial Conference on Innovative Data Systems Research* (2015).

[2] Gupta, A., Agarwal, D., Tan, D., Kulesza, J., Pathak, R., Stefani, S., and Srinivasan, V. Amazon redshift and the case for simpler data warehouses. In *Proceedings of the 2015 ACM SIGMOD international conference on management of data* (2015), ACM, pp. 1917–1923.

[3] Hausenblas, M., and Nadeau, J. Apache drill: interactive ad-hoc analysis at scale. *Big Data 1*, 2 (2013), 100–104.

[4] Kestelyn, J. Introducing parquet: Efficient columnar storage for apache hadoop. *Cloudera Blog 3* (2013).

[5] Melnik, S., Gubarev, A., Long, J. J., Romer, G., Shivakumar, S., Tolton, M., and Vassilakis, T. Dremel: interactive analysis of web-scale datasets. *Proceedings of the VLDB Endowment 3*, 1-2 (2010), 330–339.

[6] Traverso, M. Presto: Interacting with petabytes of data at facebook. *Retrieved February 4* (2013), 2014.