Farmer项目

元数据管理

HBase

Filter

Coprocessor

权限检查

语法检查

SQL Plan

SQL解析

JDBC

Farmer

Server

Thrift

1. 架构

HBase

Farmer

Server

客户端

HBase作为NoSql数据库的代表之一发展良好，其易扩展，支持高并发读写，自动分片的特性大大简化了工作，已经有很多应用使用HBase存储大数据；同时HBase也暴露出一些缺点，比如不支持SQL查询，不支持辅助索引，不支持跨行跨表事务等。Farmer将NoSql数据库的可扩展性和传统关系型数据库融合在一起。

Farmer是参照Google Megastore的开源实现，同时借鉴传统关系型数据库架构设计。Farmer是一个C/S架构的程序。

1. 架构要点

客户端：有两种方式可以连接到farmer server，1.JDBC 2.restful

1. jdbc

jdbc url格式jdbc:hbase://host:port

默认端口：10004

不支持任何额外连接属性

支持用户名和密码连接验证

支持statement和preparedStatement

结果通过ResultSet获取

由于后面接的是HBase，所以JDBC许多规范由于HBase自身的限制限制是无法实现的，对JDBC接口的实现随着功能的完备将逐步完善。

JDBC到Farmer Server的连接使用Thrift.

HBaseDriver、HBaseConnection、HBasePreparedStatement、HBaseResultSet、HBaseStatement、HBaseCallableStatement

1. restful

采用spring接受来自应用的SQL语句，返回XML或JSON格式。

客户端向服务端发送的数据：SQL语句、用户名、密码，不需要指定数据库、编码等信息，因为HBase没这些概念。

JDBC获取结果有两种模式：1数据量较小的情况下实时返回结果。2结构数据比较大的情况下服务端生成结果文件，然后ResultSet批量读取文件。

restful:1.结果实时返回 2.下载结果

两种模式可在查询之前进行配置

Farmer Server：

第一层：connectors，响应客户端连接，验证客户端身份，安全性

认证(Authenticate):连接可以是不带安全的连接，可以是带SSL的连接。

权限检查:

检查用户名、密码是否允许访问HBase，用户表存储在HBase一张单独的系统表中。

检查是否对表有操作权限(暂时不实现)

无论是JDBC连接还是rest访问都会通过该层进行验证。返回的结果统一为List，rest由rest控制端转换为JSON或XML，JDBC由JDBC客户端转换为适合于next方法的格式。

connection pool:连接池，管理连接到Farmer Server的客户端连接,

服务端通过线程来处理来自客户端的请求，服务端缓存这些客户端连接线程，不用频繁的创建或者销毁。

第二层：称为SQL Layer，有许多独立的组件构成

SQL Interface:SQL接口，接受SQL命令，并且返回用户需要查询的结果

SQL引擎：接受SQL语句，验证SQL语句合法性，解析SQL语句为对象。

sql翻译：转换SQL语句为object org.farmer.sqlparser

metadata管理：考虑到一些地方需要表的列信息，所以在内存区应该有一个表的metadata的映射

查询转换器，优化器：转换SQL语句为最优化的HBase api

org.famer.transfer

Query Cache:查询缓存，如果查询语句之前执行过并且缓存过结果则直接返回结果，这里需要结合HBase的Cache策略考虑。

管理：

管理工具 org.famer.admin

管理工具包括：Backup、Recovery、Migration、Metadata、Administration等，参考MySql等各种工具

第三层：存储引擎storage engine

这一次就是HBase本身，需要用到以下特性

过滤器：

org.farmer.filter

协处理器：

org.farmer.coprocessor

原子操作：

org.farmer.query

1. 详细架构
2. 包说明：

org.farmer.admin:管理工具包

org.farmer.jdbc:JDBC驱动

org.farmer.jdbc.metastore:元数据管理

org.farmer.sqlparser:SQL引擎

org.farmer.transfer:查询转换

org.farmer.service:farmer server服务

org.farmer.check和org.farmer.permission:权限管理

org.farmer.filter:HBase过滤器

org.farmer.coprocessor:HBase协处理器

org.farmer.query:HBase原子操作

org.farmer.exception:异常

org.farmer.utils:工具

JDBC驱动实现：

HBaseDriver:支持JDK1.7,支持用户名密码，原生的HBase是没有用户概念的，在Farmer Server中实现

HBaseConnection:只实现最常用的statement和preparedstatement,通过thrift连接到farmer server

HBaseCallableStatement:暂不支持存储过程

HBaseStatement:主要是实现execute和executeQuery

HBasePreparedStatement:

HBaseResultSet:

Farmer Server：

Connection pool:

1. Authentication:
2. 没有Connection Limits
3. Thread Reuse
4. Check Memory
5. Caches

Authentication实现：认证信息存储在HBase一张表中。由于该表非常的小，所以平时将其保存到Cache中。通过HBase API直接访问用户信息表。

SQLInterface:

接收SQL语句，组装结果返回给客户端

CommanderDispatcher：分辨出SQL语句属于哪种操作，转交给具体的模块执行，包括了SQL语句解析，这些模块包括：

查询：

数据修改包括(Insert、Update、Delete)

表修改(Create、Alert、Drop)(暂时不实现)

SQL语句解析：由jsqlparser完成，语法错误检

SQL语句转换为HBase api，执行，返回结果，被转换的语句形成执行计划。

where子句解析:jsqlparser解析出来的where条件用树表示,特点是AND和OR表达式下一定有子节点，非AND和OR表达式下一定没有子节点，采用后序遍历先访问到普通表达式，然后还是普通表达式，最后是它们之间的逻辑关系，遍历后可以组成两两节点的逻辑关系，交给filter处理

Metadata管理

SQL规范：

我们要实现的至少包括

DDL数据定义语言(CREATE,ALTER,DROP,DECLARE)

DML数据操纵语言(SELECT,DELETE,UPDATE,INSERT)

DCL 数据控制语言(GRANT,REVOKE,COMMIT,ROLLBACK)

Select语句比较复杂，目前我们考虑的主要子句：

FROM：

WHERE：

操作符：=,<>,<,>,<=,>= ,LIKE(前缀),in

AND OR

DISTINCT

GROUP BY

Alias(AS)

聚集函数:count,sum,max,min,avg

支持分页