干货丨基于Phoenix的SQL On HBase

原创 2017-05-22 王爱军 中兴开发者社区

导读

老王是技术认证教练，也是具有10多年开发经验的老码农，一直奋战在代码一线。在工作中不忘总结分享，带着大家共同进步，今天他为我们带来的文章阐述了APM产品开发中涉及到存储的关键技术，希望大家都能从中有所收获。

作者的话：

APM产品的大数据存储采用HBase，随着业务的发展，提出了支持数据聚合的需求。原生的HBase不支持聚合，如果实现聚合功能，需要自己实现聚合函数，开发周期长，且质量很难得到保证。本文主要分享下借助Phoenix，通过JDBC+SQL的方式快速、高效的对大数据进行聚合。

1. HBase介绍

HBase是一个高可靠性、高性能、面向列、可伸缩的分布式存储系统，利用HBase技术可在廉价PC Server上搭建起大规模结构化存储集群。HBase是Google Bigtable的开源实现。

1) HBase的优点

i. 列的可以动态增加，并且列为空就不存储数据，节省存储空间

ii. HBase自动切分数据，使得数据存储自动具有水平scalability

ii. HBase可以提供高并发读写操作的支持

2) 缺点

i. 不能支持条件查询，只支持按照Row key来查询

ii. 暂时不能支持Master server的故障切换

官网位置：http://hbase.apache.org/

2. Phoenix介绍

Phoenix是一个HBase的开源SQL引擎。可以使用标准的JDBC API代替HBase客户端API来创建表，插入数据，查询HBase数据。 Phoenix是构建在HBase之上的SQL引擎。通过测试，Phoenix不会降低HBase的效率，相同的功能比通过HBase的api实现会少写很多代码，且具有更好的性能。

官网位置：http://phoenix.apache.org/

3. 使用Phoenix终端访问HBase

1) 下载HBase
http://archive.apache.org/dist/hbase/hbase-1.0.3/hbase-1.0.3-bin.tar.gz
#tar -xvf hbase-1.0.3-bin.tar.gz

2)下载Phoenix
http://apache.fayea.com/phoenix/apache-phoenix-4.8.2-HBase-1.0/bin/apache-phoenix-4.

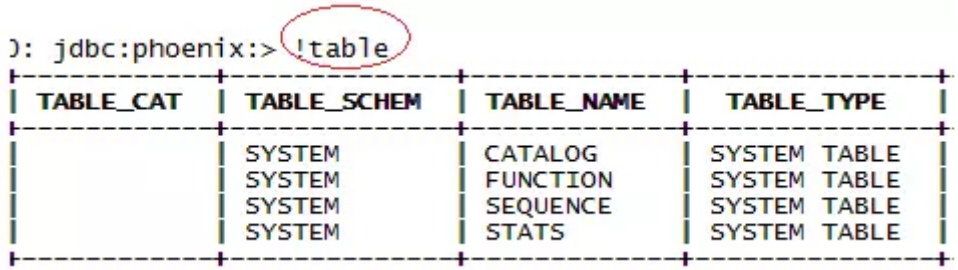
8.2-HBase-1.0-bin.tar.gz

tar -xvf apache-phoenix-4.8.2-HBase-1.0-bin.tar.gz

3)拷贝jar包
把phoenix-core-4.8.2-HBase-1.0.jar 和phoenix-4.8.2-HBase-1.0-server.jar拷贝到hbase的lib目录，启动HBase。

4) 连接Hbase
#./ apache-phoenix-4.8.2-HBase-1.0-bin/bin/sqlline.py

5) 查看表



6) 创建表
在终端执行如下Sql:
/*物理内存指标存储表*/
CREATE TABLE IF NOT EXISTS SMARTSIGHT.MEMORY (
AgentId varchar , /*采集代理标识*/
AgentStartTime varchar, /*采集代理启动时间*/
CollectTime varchar, /*指标采集时间*/

```
 PhyTotal BIGINT,       /*物理内存总和*/
 PhyFree BIGINT,        /*空闲的物理内存*/
 PhyUsed BIGINT         /*使用的物理内存*/
 /*主键*/
 CONSTRAINT memory_pk PRIMARY KEY (AgentId, AgentStartTime,CollectTime)
);
```

7) 增加或修改记录

在终端执行如下Sql:

```
 upsert into SMARTSIGHT.MEMORY (
 AgentId, AgentStartTime, CollectTime,PhyTotal, PhyFree, PhyUsed)
values ( 'FM-adaptor88','1488158514700', '20170214010505', 100,20,80);

upsert into SMARTSIGHT.MEMORY (
 AgentId, AgentStartTime, CollectTime,PhyTotal, PhyFree, PhyUsed)
values ( 'FM-adaptor88','1488158514700', '20170214010510', 200,20,180);
```

8) 查询记录

```
0: jdbc:phoenix:> select * from smartsight.memory;
+---------------+-------------------+-----------------+-----------+----------+----------+
|    AGENTID    | AGENTSTARTTIME    |   COLLECTTIME   | PHYTOTAL  | PHYFREE  | PHYUSED  |
+---------------+-------------------+-----------------+-----------+----------+----------+
| FM-adaptor88  | 1488158514700     | 20170214010505  | 100       | 20       | 80       |
| FM-adaptor88  | 1488158514700     | 20170214010510  | 200       | 20       | 180      |
+---------------+-------------------+-----------------+-----------+----------+----------+
2 rows selected (0.049 seconds)
0: jdbc:phoenix:>
```

9) 聚合查询

```
> select AgentId, AgentStartTime, TRUNC(TO_DATE(CollectTime,'yyyyMMddHHmmss'),'MINUTE') as CollectTime1
> ,max(PhyTotal) as PhyTotal, min(PhyFree) as PhyFree, sum(PhyUsed) as PhyUsed
> from SMARTSIGHT.MEMORY
> where CollectTime>='20170214010505'
> group by AgentId, AgentStartTime, CollectTime1;
+---------------+-------------------+-------------------------+-----------+----------+----------+
|    AGENTID    | AGENTSTARTTIME    |       COLLECTTIME1      | PHYTOTAL  | PHYFREE  | PHYUSED  |
+---------------+-------------------+-------------------------+-----------+----------+----------+
| FM-adaptor88  | 1488158514700     | 2017-02-14 01:05:00.000 | 200       | 20       | 260      |
+---------------+-------------------+-------------------------+-----------+----------+----------+
1 row selected (0.039 seconds)
```

4. 使用JDBC访问HBase

1) java工程pom.xml增加依赖

```
<dependency>
    <groupId>org.apache.phoenix</groupId>
    <artifactId>phoenix-core</artifactId>
    <version>4.8.1-HBase-1.0</version>
</dependency>
```

2) 代码实现

```java
public static boolean execute(String sql) {
    Connection conn = null;
    Statement stmt = null;
    try {
        conn = DriverManager.getConnection("jdbc:phoenix:127.0.0.1:2181");
        stmt = conn.createStatement();
        stmt.executeUpdate(sql);
        conn.commit();
    } catch (SQLException e) {
        return false;
    } finally {
        release(stmt, conn);
    }

    return true;
}
```

建立连接

创建Statement

执行sql

事务提交

资源释放

5. 总结

在APM产品中实现聚合功能，开始是通过java代码从HBase获取数据，然后通过计算得到聚合数据，该方式导致编写大量代码，同时网络传输数据量巨大，效率很低。

后来采用Phoenix计算聚合计算，直接在HBase中通过协处理器的方式得到计算结果，然后通过网络把结果返回给调用方，同时采用了sql的方式实现，开发效率极大提升。