使用 Hive 构建数据仓库

2013-8-19 10:16 | 发布者: joanne800 | 查看: 667 | 评论: 0

摘要: 人们声称 Hive 是 Hadoop 的数据仓库。尽管从某个层面上讲这是真的，但这种说法也有几分虚假。不过，有时您必须使用您可以使用的工具，就此而言，可以将 Hive 用作一个数据仓库。

有 3 个家伙来到了一家企业。第一个（数据仓库）身材魁梧：他带来了历史和经验，而且能言会道，所说的大部分话都是真的。但是，在许多方面，它有些自我膨胀，在另一些方面又有些铺张浪费，而且人们厌烦了各种结果的代价。Apache Hadoop 进入了同一栋建筑，声称要接管整个市场。他大肆鼓吹大数据、速度、数据量、种类以及一堆 v 开头的词汇，这些词汇在市场营销计划之外没有多大意义。他漫不经心地说着分析、预测等等。而且他要价很低。于是人们开始停下来倾听。

Apache Hive 在屋外徘徊，他没有打算和其他人争论。他希望与 Hadoop 合作，但不同于 Hadoop，他不希望将数据仓库抛在路边。Hive 拥有数据仓库功能，但在商业智能 (BI) 和分析上有一些限制。它具有数据库的潜力，但也具有关系数据库管理系统 (RDBMS) 和结构化查询语言 (SQL) 方面的限制。它更加开放和诚实。它与数据仓库密切相关，与 RDBMS 也密切相关。但它从未站出来声称它并不像初看起来那么简单。Hadoop 打断了谈话，声称它是 Hadoop 领域的数据仓库。Hadoop 似乎让出了最优秀营销公关代表的地位，在一次简单的对话之后，结果变成了是 Hive 和 Hadoop 在拯救世界。这种描述很吸引人，也很有趣。但它是真的吗？ 有几分相似。

数据仓库

构建一个真正的数据仓库可能是一个庞大的工程。有许多不同的设备、方法和理论。最大的共同价值是什么？事实是什么，哪些主题与这些事实相关？以及您如何混合、匹配、合并和集成可能已存在数十年的系统与仅在几个月前实现的系统？这还是在大数据和 Hadoop 之前。将非结构化、数据、NoSQL 和 Hadoop 添加到组合中，您很快就会得到一个庞大的数据集成项目。

描述一个数据仓库的最简单方式是，认识到可以将它归结为星形模式、事实和维度。您如何创建这些元素，决定权在您手上 — 通过暂存数据库；动态提取、转换、加载流程；或者集成辅助索引。当然，您可以构建一个包含星形模式、事实和维度的数据仓库，使用 Hive 作为核心技术，但这并不容易。在 Hadoop 世界外部，这会成为一个更大的挑战。与其说 Hive 是一种合法的数据仓库，倒不如说它是一个集成、转换、快速查找工具。该模式可能像是数据仓库，但适用性表明它不是 RDBMS。那么为什么使用它？

简单来讲，有时您需要使用摆在面前的工具。

任何从事过一段时间的 IT 工作的人都可能告诉您，适合一项工作的正确工具并不总是能够用到。或者，正确的工具虽然用得到，但为了削减成本会阻碍使用该工具。有时企业政治学发挥着重大作用。无论什么原因，我们大部分人在很多情形下被迫使用可能并不是最适合其工作的工具来构建、设计和开发。

在我参与的许多项目中，我不得不使用 Hive 作为数据库、作为数据仓库以及作为缓慢变化的系统。这很有挑战性，但偶尔会令人生厌。有时，您不得不摇头并想知道为什么。但在一天结束时，您仍然需要让它工作。如果需要在 Hive 中构建和使用某个数据仓库，而且需要使用缓慢变化的维度和更新，并协调旧数据，那么您必须这么做。重点并不总是提供最佳的工具，而是创建最适合您工作的工具。

星形模式是什么

想象一颗星星 — 具有一个中心和多个指向不同方向的 “手臂”。中心是动力之源或事实表。所有手臂都指向不同维度。许多数据仓库有一个事实表和多个维度。

事实表包含您可以加权或计算的任何数据。在此示例中，您拥有棒球统计数据，比如跑垒、全垒打、击球率等。您可以计算、增加、减去或乘以这些列。

维度更加以主题为基础。在此示例中，您有运动员信息维度、时间和日期维度，等等。通常没有计算或加权多个维度中的列。

在此示例中，将一个维度表与一个事实表连接的键是 playerID。

Hive

由于 Hive 的类 SQL 功能和类数据库功能，它向非编程人员开放了大数据 Hadoop 生态系统。它常被描述为一个构建于 Hadoop 之上的数据仓库基础架构。这是一种部分真实的表述（因为您可将源数据转换为星形模式），但在创建事实表和维度表时，它更关乎设计而不是技术。

尽管如此，Hive 并不真正是一个数据仓库。它甚至并不真正是一个数据库。您可以使用 Hive 构建和设计一个数据仓库，也可以使用 Hive 构建和设计数据库表，但存在的一些限制需要提供许多解决办法，并且将会带来一些挑战。

例如，索引在 Hive 中有一些限制。如何克服这个问题呢？您可以使用`org.apache.hadoop.hive.ql.index.compact.CompactIndexHandler`函数在 Hive 中创建索引。Hive 和缓慢变化的维度并不总是可能实现。但是如果构建暂存表和使用一定量的连接（而且计划添加一个新表，转储旧表，并且只保留最新、更新表用于比较），则可能实现它们。

连接到 Hive 的外部报告或分析系统是一个巨大的问题。甚至对于 JDBC 连接，也仅限于连接到默认数据库。人们在寻求更多的经过改进的元数据，而且 Apache HCatalog 等工具正在帮助将各种服务连接到 Hive 元存储。在未来，如果利用得当，这可能是一个重大的增值区域。

所以，尽管 Hive 不是一个可靠的数据仓库或数据库，但仍然可以使用一些方法将 Hive 用作数据仓库或数据库。只是需要做一些工作和利用一些解决办法将 Hive 打造成这样的系统。为什么您要再次经历这一过程？因为您必须使用手头的工具并让它们发挥作用。

示例：为棒球信息构建一个数据仓库

下面的棒球数据示例展示了如何在 Hive 中使用来自 Sean Lahman 网站的棒球数据设计和构建一个数据仓库。我很喜欢挑战数据仓库的反规范化 (denormalizing) 并从该数据构建一个数据仓库。在 “使用 Hive 为数据构建一个库” 中，我使用 VMware Fusion 在我的 Macbook 上创建了一个 IBM InfoSphere® BigInsights™ 虚拟机 (VM)。这是一个简单测试，所以我的 VM 只有 1 GB RAM 和 20 GB 固态硬盘存储空间。操作系统是 Linux® 的 CentOS 6.4 64 位发行版。

要开始使用此示例，请下载 IBM InfoSphere BigInsights Basic Edition（参见 参考资料）。您需要有一个 IBM Universal ID 或注册获取一个 ID，然后才能下载 InfoSphere BigInsights Basic Edition。

InfoSphere BigInsights Quick Start Edition

InfoSphere BigInsights Quick Start Edition 是 InfoSphere BigInsights（IBM 的基于 Hadoop 的产品）的一个免费的可下载版本。使用 Quick Start Edition，您可以尝试使用 IBM 开发的特性来提高开源 Hadoop 的价值，比如 Big SQL、文本分析和 BigSheets。引导式学习可让您的体验尽可能地顺畅，包括按部就班、自订进度的教程和视频，可帮助您开始让 Hadoop 为您所用。没有时间或数据限制，您可以自行安排时间，在大量数据上试验。请 观看视频、学习教程 (PDF) 和 下载 BigInsights Quick Start Edition。

导入数据

首先下载包含棒球和棒球运动员的统计数据的 CSV 文件（参见 下载）。在 Linux 内创建一个目录，然后运行：

```
$ Sudo mkdir /user/baseball.


sudo wget http://seanlahman.com/files/database/lahman2012-csv.zip
```

相关分类

数据集成 大数据
数据仓库 商业智能
数据挖掘

下级分类

ODS Data Warehouse
Data Mart Data Model

示例包含 4 个主要表，每个表有一个惟一列（Master 表、Batting、Pitching 和 Fielding）以及多个次要表。

设计数据仓库

此数据对一个数据库而言是结构化数据，但对于数据仓库，您需要找出事实和维度。数据仓库设计很简单：您对该数据库进行反规范化，基于运动员统计数据创建一个事实表。然后基于与这些统计数据相关的某些主题区域来创建维度。在连接方面，Hive 表现不是很好，而 MapReduce 也好不了多少，所以拥有一个反规范化的星形模式对某些查询会有所帮助。

设计包含一个名为 fact_Player_Stats 的事实表，它包含各种 CSV 文件和表中包含的每个统计列。您需要使用来自核心表（Batting、Pitching 和 Fielding）的数据，以及来自一些补充表（它们也包含统计数据）的数据。因此，必须添加来自以下表的统计列：

- AllStarFull
- hall of Fame
- BattingPost
- PitchingPost
- FieldingOF
- Salaries
- AwardsPlayers
- AwardsSharePlayers
- Appearances
- SchoolsPlayers

一些表仅包含少数统计列。例如，在 FieldingOF 表中，您只需要将列 stint、Glf、Gcf 和 Grf 添加到 fact_Player_Stats 事实表。对于 SchoolsPlayers 表，只需获取 yearMin 和 yearMax 列。对其他表采取类似的步骤。事实表中仅需要统计列。

备注：您不会使用来自 Managers、Teams、TeamsHalf、SeriesPost 等表的任何数据。

fact_Player_Stats 事实表仅包含键 playerID、FranchID、yearID 和 SchoolID。对于维度表，您必须去掉统计数据（如果存在），仅保留与主题相关的列：

- dim_Players 维度表从 Master 表中获取数据（运动员姓名、生日、传记信息）。主键为 playerID。
- dim_TeamFranchise 维度表从 TeamFranchise 表中获取所有数据。主键为 FranchID。
- dim_Schools 维度表从 Schools 表中获取所有数据。
- dim_Year 是一个基于月份和年份的时间维度表 (1871-2012)。

将数据库用于数据仓库

如果尚未创建棒球数据库，推荐您立即这么做，然后根据这些基础表来构建数据仓库。可通过编写复杂的脚本，从一个平面文件构建数据仓库，然后将同一个平面文件重用于另一个表，但对于本文，我选择使用之前在 “使用 Hive 为数据构建一个库” 中创建的数据库。

使用 Hive 构建数据仓库

完成数据分析和设计之后，是时候基于您的星形模式设计来构建数据仓库了。在 Hive shell 中，创建 baseball_stats 数据库，创建表，加载表，验证这些表是否正确。（此过程已在 “使用 Hive 为数据构建一个库” 中提供。）接下来，创建数据仓库事实表。清单 1 给出了相关代码。

清单 1. 创建数据仓库事实表

```
$ Hive

Create Database baseball_stats;

Create table baseball_stats.fact_player_stats as
         ( SELECT a.playerID, FranchID, yearID, SchoolID, stint int, g int,
g_batting int, ab int, r int, h int, 2b int, 3b int, hr int, rbi int, sb int,
cs int, bb int, so int, ibb int, hbp int, sh int, sf int, gidp int, w int,
l int, g int, gs int, cg int, sho int, sv int, ipouts int, ph int, er int,
phr int, pbb int, pso int, baopp int, era int,pibb int, wp int, phbp int,
bk int, bfp int,gf int, pr int, p sh int, psf int, p gidp int, fg int,
fgs int, innouts int, po int, a int, e int, dp int, pb int, wp int, fsb int,
fcs int, zr int, gamenum int, allstargp int, ballots int, needed int,votes int,
playoff_g int,playoff_ab int, playoff_r int, playoff_h int,
playoff_2b int, playoff_3b int, playoff_hr int, playoff_rbi int, playoff_sb int,
playoff_cs int, playoff_bb int, playoff_so int, playoff_ibb int, playoff_hbp int,
playoff_sh int, playoff_sh, playoff_sf int, playoff_gidp int, pitchplayoff_w int,
pitchplayoff_l int, pitchplayoff_g int, pitchplayoff_gs int, pitchplayoff_ cg int,
pitchplayoff_sho int, pitchplayoff_sv int, pitchplayoff_ipouts int,
pitchplayoff_h int,pitchplayoff_er int, pitchplayoff_hr int, pitchplayoff_bb int,
pitchplayoff_so int, pitchplayoff_baopp int, pitchplayoff_era int, pitchplayoff_ibb int,
pitchplayoff_wp int,pitchplayoff_hbp int, pitchplayoff_bk int, pitchplayoff_BFP int,
pitchplayoff_gf int, pitchplayoff_r int, pitchplayoff_sh int, pitchplayoff_sf int,
pitchplayoff_gidp int, glf int, grf int, gcf int, salary double, award int,
fieldplayoffs_g int, fieldplayoffs_gs int, fieldplayoffs_innouts int,
fieldplayoffs_po int, fieldplayoffs_a int, fieldplayoffs_e int, fieldplayoffs_dp int,
fieldplayoffs_dp int,fieldplayoffs_tp int, fieldplayoffs_pb int, fieldplayoffs_sb int,
fieldplayoffs_cs int,  appearances_g_all int, appearances_gs int,
appearances_g_batting int, appearances_defense int,
appearances_g_p int, appearances_g_c int, appearances_g_1b int,
appearances_g_2b int, appearances_g_3b int, appearances_g_ss int, appearances_g_ss int,
appearances_g_lf int, appearances_g_cf int, appearances_g_rf int, appearances_dh int,
appearances_ph int, appearances_pr int, yearMin double, yearMax double
from baseball.Batting B JOIN Pitching P ON B.playerid = P.playerID
JOIN fielding F ON B.playerID = F.playerID
JOIN Team T ON b.teamid = t.teamid JOIN TeamFranchises TF ON
t.franchid = tf.franchid ...);
```

现在，创建数据仓库维度表。清单 2 给出了相关代码。

现在，创建数据仓库维度表。清单 2 给出了相关代码。

清单 2. 创建数据仓库维度表

```
$ Hive

Create table baseball_stats.dim_Players AS

        ( SELECT lahmanID int, playerID int, managerID int, hofID int, birthyear INT,
          birthMonth INT, birthDay INT, birthCountry STRING, birthState STRING,
          birthCity STRING, deathYear INT, deathMonth INT,deathDay INT,
          deathCountry STRING, deathState STRING, deathCity STRING,
          nameFirst STRING, nameLast STRING, nameNote STRING, nameGive STRING,
          nameNick STRING, weight decimal, height decimal, bats STRING,
          throws STRING, debut  INT, finalGame INT,
          college STRING, lahman40ID INT, lahman45ID INT, retroID INT,
          holtzID INT, hbrefID INT
          FROM baseball.master .... );
```

运行一个查询

我们运行一些查询来确保数据看起来很正常。首先，选择事实表中的所有数据（将其限制到前 10 行）。可运行其他两个查询来确保维度表看起来正常，确保它们已连接到事实表，等等。也可以在事实表上运行一次统计，确保总行数一致。当然，您需要将该数据关联到原始基础表并进行累加。清单 3 显示了测试数据是否存在和正确，以及维度是否连接到事实表的代码。

清单 3. 通过测试来了解俄数据是否存在和正确，以及维度是否连接到事实表

```
$ HIVE
        Use baseball_stats;
        Select * from fact_player_stats limit 10;

        Select A.PlayerID, A.name, B.teamID, B.AB, B.R, B.H, B.2B, B.3B, B.HR, B.RBI
        FROM dim_players A JOIN fact_player_stats B ON a.playerid = b.playerid;

        Select count(*) from fact_player_stats;

        Select count(*) from dim_players.

        Select max(r) from fact_player_stats where playerid=1234;
```

如果希望对 Hive 数据仓库中的数据进行更彻底的验证，可以对某些列或所有列计算最小值、最大值或平均值，并将结果与原始基础表进行比较。然后寻找准确的匹配值。

已同步至 joanne800的微博

上一篇：企业数据仓库到大数据如何过度

下一篇：数据仓库基本知识你了解多少？

最新评论

验证问答 换一个 验证码 换一个

评论