



一种关联感知的大数据导入方法

公怀予¹, 徐劲松², 王攀²

(1. 中国电信股份有限公司济源分公司, 河南 济源 454650;

2. 南京邮电大学, 江苏 南京 210003)

摘要: 针对现有数据库向大数据迁移的背景, Apache 推出了 Sqoop 作为关系数据库向大数据迁移的主要工具。Sqoop 简单地将数据表切分并随机存储到不同的节点上。针对 Hadoop 的这种存储方式带来的关系查询的低效率问题, 设计了一种关联度感知的数据导入预处理方法。将关联度较高的表尽量存储在相邻的虚拟机节点, 以降低关联数据查询带来的网络传输时延, 提高系统的性能。对比实验表明, 将关联性较强的数据表存放在相同或相邻节点上, 可以成倍提高数据查询的性能。

关键词: 大数据; Sqoop; Hadoop; NoSQL

中图分类号: TP393

文献标识码: A

doi: 10.11959/j.issn.1000-0801.2016044

An associated perception import method for big data

GONG Huaiyu¹, XU Jinsong², WANG Pan²

1. Jiyuan Branch of China Telecom Co., Ltd., Jiyuan 454650, China

2. Nanjing University of Posts and Telecommunications, Nanjing 210003, China

Abstract: Against the background of the existing database to the large data migration, Apache introduced the Sqoop as the main tool for the relational database to the big data migration. Sqoop simply cut the data table and randomly store it on different nodes. Being aimed at the problem of low efficiency of the query of the relationship between the Hadoop, a method of data importing and preprocessing was designed. To reduce the network transmission delay and improve the performance of the system, the high correlation degree was kept in the adjacent nodes. The contrast experiment shows that the performance of the data query can be improved greatly by the same or adjacent nodes.

Key words: big data, Sqoop, Hadoop, NoSQL

收稿日期: 2015-09-01; 修回日期: 2015-12-20

基金项目: 江苏省自然科学基金资助项目 (No.BK2009426); 江苏省高校自然科学基金资助项目 (No.14KJD520005); 2013 江苏省六大人才高峰计划基金资助项目; 2013 国家发展和改革委员会信息安全专项基金资助项目; 国家电网公司 2014 年科技项目: 电力信息通信网络流量预测和管理智能化关键技术研究及其应用; 2015 江苏省产学研前瞻性联合研究项目 (No.BY2015011-02)

Foundation Items: Jiangsu Province Project of Natural Science Foundation (No.BK2009426), Jiangsu Province Education Project of Natural Science Foundation (No.14KJD520005), 2013 Six Talent Peaks Project in Jiangsu Province, 2013 Information Security Special Funds of the National Development and Reform Commission, State Grid 2014 Science and Technology Project: Research and Application of Network Traffic Prediction and Smart Pipe Key Technologies for Electric Power Information Communication Network, 2015 Industrialization, Education and Research Joint Prospective Project (No.BY2015011-02)

1 引言

大数据从最初为 Google、Facebook 等公司解决大量数据的存储问题,到现在被越来越多的企业应用。目前,对于大数据的解决方案,主要有商业解决方案和 Hadoop 生态系统解决方案两种。商业解决方案具有使用方便、集成度高以及性能稳定的特点。但 Hadoop 生态系统多样、灵活、高扩展性等优势吸引了中小企业的注意并成为其主要的

大数据解决方案。大数据应用中,数据主要源于互联网和已有的结构化数据两者。其中,互联网数据催生了各种各样的大数据创新应用。而结构化数据是已有应用的保障,在我国,7 成以上的大数据来源都是结构化数据^[1]。为了实现关系型数据仓库向大数据的迁移,Apache 提供了 Sqoop^[2]工具,以协助传统的关系型数据库管理系统 (relational database management system, RDBMS) 与 Hadoop 之间进行高效的大数据交流。

实践发现,在使用 Sqoop 导入 Hadoop 的过程中,初始的设定将会把一张表切分成 4 等分,再使用 mapper 通过 JDBC 将数据导入 NoSQL (如 HDFS、HBase 等),而这些 mapper 会根据 Hadoop 的工作配置,随机地存储在被启动空间的节点上,当数据库应用中的查询语句中包含 join 语法时,需要将两个表中的相应字段进行合并,这时候数据的传输需要通过网络传输来完成。考虑到网络传输会比直接在硬盘上读取数据慢了很多,本文提出一种称为基于关联感知 Sqoop (association-based Sqoop, AB_Sqoop) 的数据导入算法,通过数据表关联性分析的方法对关系数据库进行预处理,在数据导入的过程中,将相关性较高的数据表尽量存放在同一硬盘或相邻节点,以减少数据的网络传输,加速大数据应用的整体性能。

2 相关研究

由于现在的企业环境中有着大量的结构化数据,这些数据并不能简单地使用 MapReduce 来处理,因此需要建立一种不更改复杂的 Hadoop 架构的机制来处理结构化数据^[3,4]。大多数的 NoSQL 查询语句使用 Hive 来作为 SQL 到

MapReduce 的翻译工具,但很多论文提到 Hive^[5]所翻译出的 MapReduce 并不是很有效率。参考文献[6]提出将整个查询语句根据其关联性将不必要的 join 进行合并,以减少非必要的 join 操作,大大降低了查询所需要的时间,这种做法在 2012 年被 Hive 合并。

由于 join 是数据库中常见的耗时操作,许多的研究都针对 join 操作的优化进行探索,参考文献[7]在 reduce 阶段将中间产物重新分区并复制给其他已经完成的节点上以加速运算,参考文献[8]使用 MapReduce 的简化 join 模型,参考文献[9]使用 MapReduce 模型的自然延伸加入过滤操作,使 MapReduce 先使用逻辑过滤来省略中间而让整个完成时间减少。

3 基于关联度感知的 Sqoop 导入

基于关联感知的 Sqoop 导入的流程如图 1 所示,第一步是分析数据库的日志,根据数据表之间的关联查询的数目获得数据表之间的关联性,第二步是考虑数据表的大小,计算获得数据表之间关联度的关系效用,最后查表获得数据表之间的关联度,再根据这个关联度导入 Hadoop 环境中。

3.1 分析数据表之间的关联性

关系数据库中的日志记录了数据库中的所有操作,包括数据库的设定、数据的修改以及查询。设计关系数据库的日志的目的是在数据库发生问题的时候,可以通过查询日志的方式解决问题。

由于关系数据库的日志记录了数据库查询的记录,因此发生的关联查询也会在日志中体现,因此,分析数据库日志可以获得关系数据库之间表的关联性,比如表 1 与表 2 之间产生的关联查询为 15 次,则该两表之间的关联性记为 15。且记表 n 与表 n 之间的关联性为 0。将这些关联性通过一个二维表可以将其列出,记该表为 table association (TA)。

3.2 数据表之间的大小

除了考虑表之间的关联性以外,还需要考虑数据表的大小问题。这是因为当两张表的关联性非常大,但其大小相对于数据库中其他的表很小的情况下,将两张表放在一

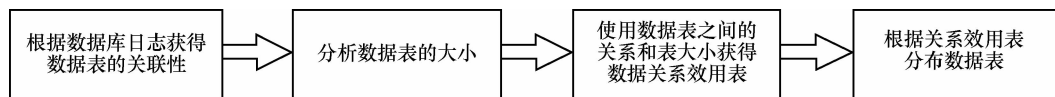


图 1 基于关联感知的 Sqoop 导入基本流程



起与否,并不能对关联查询的性能起到很好的作用。这是因为,Hadoop 在执行任务的时候,会在数据量比较大的节点上发起任务,通过这样的手段减少网络传输,从而提高执行的效率。

同样,将数据库中的表两两比较,选出较小的表作为代表,形成一个二维列表 TS(table size)。如表 1 的大小是 10 MB,而表 2 的大小是 30 MB,则表 1 行与表 2 列的数据记录为 10,同样,记表 n 与表 n 之间的大小记录为 0。

3.3 关系效用的计算

关系效用(effective association)使用 TA 和 TS 对应的位置两两相乘即可得出,即求 TA 与 TS 的点积:

$$ECT=TA \cdot TS \quad (1)$$

从 TA 与 TS 的生成可以看出, $TA^T=TA$,且 $TS^T=TS$,由式(1)同样可以得出 $ECT^T=ECT$ 。

由于 ECT 是一个对称矩阵,因此可以从主对角线取其上半部分,也就是说将 ECT 的下半部分都记为 0,得到 ECT' 。实际上由于 TA 与 TS 都对称,在计算之初就可以将其下半部分取 0,得到 TA' 和 TS' 。即:

$$ECT'=TA' \cdot TS' \quad (2)$$

3.4 Sqoop 导入

考虑到需要将关系效用最大的表尽量存放到一起,因此,首先需要明确每个虚拟机(VM)能存放的表的分片(table split)的大小,该容量限制记为 CA (capacity),则导入的流程如下:

步骤 1 将 ECT' 中的每行取 $CA-1$ 个较大的值相加为 St ;

步骤 2 取 St 中最大的为 $Smax$,并首先将 $Smax$ 组成的表首先导入,并将 $Smax$ 组成的表标示为已导入;

步骤 3 如果剩余表的数量大于 CA ,则重复步骤 1、步骤 2;

步骤 4 将剩余的表做最后一次导入。

为了验证算法的有效性,在步骤 2 中引入变量数据位置(data locality,DL)。在这一步骤的导入过程由于数据存放在相同节点,因此可以理解为这些数据的关联查询不需要网络传输。假设 $Smax$ 的组成为第 n 行上的 $i,j,k,\dots$ 列,则该 DL 的值为:

$$DL=ECT'_{[ni]}+ECT'_{[nj]}+ECT'_{[nk]}+\dots+ECT'_{[ij]}+ECT'_{[ik]}+\dots+ECT'_{[jk]}+\dots=S_{max}+ECT'_{[ij]}+ECT'_{[ik]}+\dots+ECT'_{[jk]}+\dots \quad (3)$$

由式(3)可知, DL 的值包括了所有需要标记为已导入的表的效用关系量,该值累加得到最后的 DL 值。当 DL 值

越大,则意味着所需要的网络传输数据越小。

Sqoop 导入的算法描述如下。

算法 基于感知度的 Sqoop 导入算法

输入 ECT' :关系效用表

n :数据库表的数量

CA :每一个节点可以存储的数据表分片的数量

$DL=0$:初始化 DL 值

$P=1$

输出 数据表位置的集合 $\{S_1, S_2, \dots, S_\beta\}$, $\beta=n/CA$

$\alpha=CA-1$

while ($n>0$) {

for ($i=1; i<n; i++$) {

select α values form row $_i$ of ECT' , 假设 $ECT'_{[im_1]} \dots$

$ECT'_{[im_\alpha]}$

$Sum_{row_i} = \sum_{j=1}^{\alpha} ECT'_{[im_j]} / m_j$ 是 ECT' 上最大值的列数

}

$Sum_{max} = \max\{Srow_1, Srow_2, \dots, Srow_n\}$, 假设 $Sum_{max} = Srow_k$

从 ECT' 移除行 $k, m_1, m_2, m_3, \dots, m_\alpha$

从 ECT' 移除列 $k, m_1, m_2, m_3, \dots, m_\alpha$

将表 $t_k, t_{m_1}, t_{m_2}, t_{m_3}, \dots, t_{m_\alpha}$ 加入集合 Sp

$P++$

$L+=Sum_{max}$

$n-=CA$

}

由以上算法可知,该算法拆解部分的时间复杂度为 n^2 ,而总共需要完成 n/CA 次循环,则总的时间复杂度为 n^3/CA 。

4 实验与对比分析

4.1 模拟实验

模拟实验主要以原始的 Sqoop 与使用了基于关联感知的 Sqoop 对 DL 值进行模拟比较。

首先定义模拟实验的相关数据如下。

- 数据表大小:1~10 GB,指一个表的随机大小值。
- 数据表关联性:1~1 000,指模拟数据库日志中的关联查询书目。
- 节点容量:2~50,指节点能够存储数据表分片的上限。

- 表的数目:20~300,指数据库中表的数量。
- 节点数目:20~100,指数据节点的数量。

如图 2 所示,当取数据表的数目为 100、节点容量为 20、表的大小取随机 1~10 GB、关联性取随机 1~500 的时候,随着节点数目的变化,其 DL 值的变化。

从图 2 可知,当数据节点很少的时候,使用关联感知导入的效率并不占优,但随着节点数目的变大,该算法的 DL 值的变化比较稳定,而原始 Sqoop 会随着节点数目变大,其 DL 值不断变小。从第 2.4 节的分析可知,由于数据随机存放在不同节点,因此,数据查询的性能不断降低,到了节点数为 100 的时候,关联感知算法的性能达到了传统算法的 397%。

4.2 实测比较

实测环境使用 Apache Ambari 工具在实验室局域网环境下搭建一个 Hadoop 集群,15 台 PC,其中 1 台为 master 节点,14 台为 slave 节点。安装 Hive、MapReduce 和 Sqoop。为了凸显网络传输对性能的影响,将 PC 的网络传输调为 10 Mbit/s 共享。使用苏州锐创公司的运营 Oracle 数据库作为导入的数据源,该数据源数据 241 MB,分别进行传统的 Sqoop 导入和 AB_Sqoop 导入,使用 join 操作进行关联查询,其实验结果如图 3 所示。

从图 3 可知,当数据查询参与的数据表变多的时候,查询的时间相对增大,而 AB_Sqoop 基本维持在 Sqoop 的一半时间,即使用 AB_Sqoop 能提高一半以上的性能。

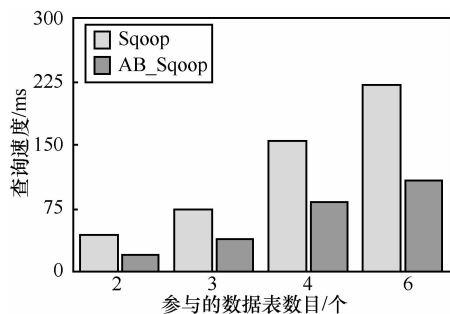


图 3 join 操作的时间

5 结束语

基于 Hadoop 构建大数据的分析平台是目前中小企业的共同选择。在传统的关系数据库导入大数据平台时,虽然 Apache 提供了 Sqoop 工具供数据的导入导出,但由于设计思想和数据查询的方式不同,传统的关系查询应用会产生不利的影响,特别是 join 操作会激起消耗运算资源。本文的分析和实验证明:将关联性较大的数据表存放到相同节点或相邻节点的做法可以有效地避免网络传输带来的性能影响。下一步需要进行的工作是考虑 HDFS 中数据副本对导入 Hadoop 后性能的影响。此外算法中节点容量上限的定义虽然简化了节点存储的问题,但实际情况中导入数据库的过程不可能有一个空置的大数据平台给用户使用,需要更进一步考虑每个节点拥有的容量。

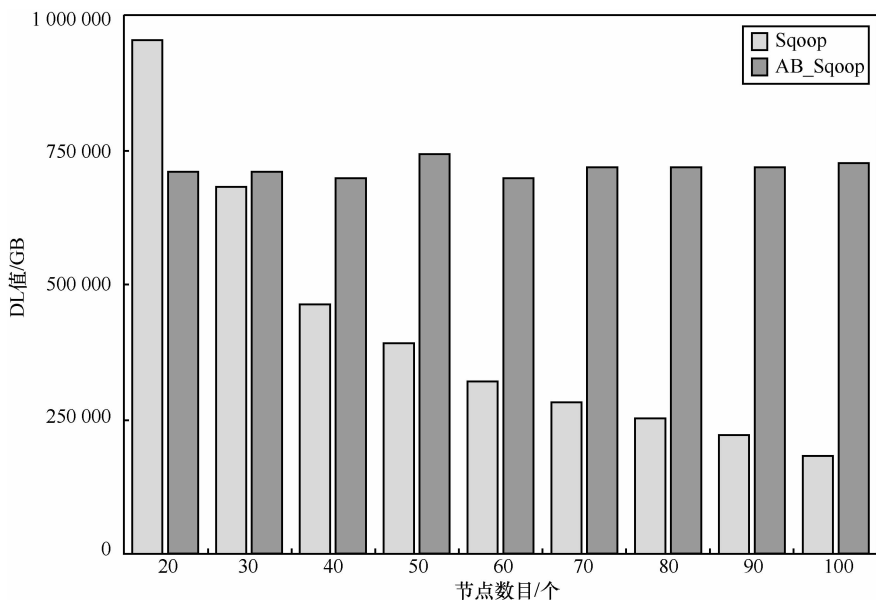


图 2 不同节点数的数据距离



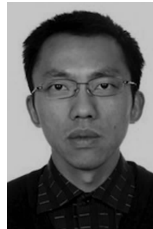
参考文献:

- [1] 中国大数据发展调查研究结果 [EB/OL]. (2015-07-29) [2015-09-01]. <http://zhishi.moojnn.com/?/article/262>.
Research report of China big data development [EB/OL]. (2015-07-29)[2015-09-01]. <http://zhishi.moojnn.com/?/article/262>.
- [2] Apache Sqoop[EB/OL].[2015-09-01]. <http://sqoop.apache.org/>.
- [3] BALMIN A, KALDEWEY T, TATA S. Clydesdale: structured data processing on Hadoop [C] // 2012 ACM SIGMOD International Conference on Management of Data, May 20-24, 2012, Scottsdale, AZ, USA. New York: ACM Press, 2012: 705-708.
- [4] BALMIN A, KALDEWEY T, TATA S. Clydesdale: structured data processing on MapReduce [C] // 2012 International Conference on Extending Database Technology, March 27-30, 2012, Berlin, German. New York: ACM Press, 2012: 15-25.
- [5] THUSOO A, SARMA J S, JAIN N, et al. Hive - a warehousing solution over a MapReduce framework [J]. PVLDB, 2009, 2(2): 1626-1629.
- [6] LEE R B, LUO T, HUAI Y, et al. YSmart: yet another SQL-to-MapReduce translator[C]//2011 International Conference on Distributed Computing Systems, June 20-24, 2011, Minneapolis, Minnesota, USA. New Jersey: IEEE Press, 2011: 25-36.
- [7] LYNDEN S, TANIMURA Y, KOJIMAL, et al. Dynamic data redistribution for MapReduce joins[C]//2011 IEEE International Conference on Cloud Computing Technology and Science, November 29-December 1, 2011, Athens, Greece. New Jersey: IEEE Press, 2011: 717-723.
- [8] ALPER O, MIREK R. Processing theta-joins using MapReduce[C]// 2011 ACM SIGMOD International Conference on Management of Data, June 12-16, 2011, Athens, Greece. New Jersey: IEEE Press, 2011: 949-960.
- [9] JIANG D W, TUNG A K H, CHEN G. Map-join-reduce: toward scalable and efficient data analysis on large clusters [J]. IEEE Transactions on Knowledge and Data Engineering, 2011, 23(9): 1299-1311.

[作者简介]



公怀予(1973-),男,中国电信股份有限公司济源分公司工程师、总经理,主要研究方向为大数据分析和流量经营。



徐劲松(1974-),男,博士,南京邮电大学通达学院副教授、教研室主任,主要研究方向为信息安全、云计算及大数据应用。



王攀(1979-),男,南京邮电大学副研究员,主要研究方向为大数据分析和流量经营。