基于 NoSQL 系统的组合索引技术研究

宗 平[1],吴秀娟[2]
(1. 南京邮电大学 海外教育学院,江苏 南京 210023;
2. 南京邮电大学 计算机学院,江苏 南京 210023)

摘 要:分布式 NoSQL 系统旨在提供大规模数据的高可用性,但缺乏内在的支持复杂查询的应用程序。传统的基于单一词汇倒排表的解决方案未达到良好的效果。因此,文中就文档型数据库在处理动态文档集时不支持多键作为主索引的缺点展开研究,提出了一种改进的组合索引方法。通过存储组合条件的倒列表,查询驱动机制可以从最近的查询记录中自适应地存储比较受欢迎的条件组合。该方法可以降低整体的带宽消耗,只需占用较少的存储资源等额外开销,明显改善了 NoSQL 系统的容量和响应时间。

关键词:NoSQL;组合索引;数据查询
中图分类号:TP31 **文献标识码**:A **文章编号**:1673-629X(2014)12-0053-04
doi:10.3969/j.issn.1673-629X.2014.12.013

Study of Multiterm Indexing Techniques Based on NoSQL System

ZONG Ping[1], WU Xiu-juan[2]
(1. College of Overseas Education, Nanjing University of Posts and Telecommunications, Nanjing 210023, China;
2. College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210023, China)

Abstract: The purpose of distributed NoSQL systems is to provide high availability for large-scale of data, but they are short of the inherent support for complex queries that often required by overlying applications. The traditional solutions based on inverted lists for single terms perform are poorly in large-scale distributed settings. Hence, research the shortcoming that document database does not support multiple key when dealing with dynamic set of documents as the primary index, then propose an improved multiterm indexing technique. By storing the inverted lists of combinations of terms, a query-driven mechanism adaptively stores the popular term combinations derived the recent query history. This approach reduces the overall bandwidth consumption, only marginal overhead in terms of additional, but few required storage resources, obviously improving the NoSQL system's capacity and response time.

Key words: NoSQL; multiterm index; data query

0 引 言

大规模分布式、高度可用的存储系统的迫切需要促使了 NoSQL(Not only SQL)系统的兴起。NoSQL 是一种不同于传统的关系型数据库的数据库管理系统的统称[1]。为了实现数据的高可用性,NoSQL 系统摒弃了很多作为关系数据库管理系统的标准功能,比如强大的数据一致性、数据的访问控制、标准化的查询语言和参照完整性约束。但是 NoSQL 系统查询功能却是有限的。

倒排文件是全文检索中广泛使用的索引结构,对静态文档集合建立倒排索引的研究已有较长时间。倒排索引是信息检索系统的核心部分,其存储结构对检索的效率和效果起着至关重要的作用[2]。随着计算机技术的发展,需要存储的数据越来越大。同时特定的应用领域如新闻搜索、桌面搜索等对实时更新性能要求较高,这需要使用有效的索引更新策略,也称动态索引[3]。在动态文档环境下,如标签系统,用户可以添加或删除标签或文件。但这些动态特性导致计算多个键

收稿日期:2014-01-02 **修回日期**:2014-04-09 **网络出版时间**:2014-09-11
基金项目:国家科技重大专项(2011ZX03005-004-03)
作者简介:宗 平(1956-),男,江苏南京人,教授,硕士生导师,CCF 会员,研究方向为计算机网络技术、智能数据处理技术、物联网技术、软件工程等;吴秀娟(1986-),女,山东莱芜人,硕士研究生,研究方向为并行分布式计算。
网络出版地址:http://www.cnki.net/kcms/detail/61.1450.TP.20140911.1010.049.html

作为主索引的有意义数据是不易处理的。因此,提出一种查询驱动优化技术,仅存储被经常查询的数据的键。为了有效处理不断变化的数据对象,采用增量更新机制。显然在存在更新时,要有一个用于处理查询和维护索引的成本权衡。文中的研究结果显示,组合索引能使整体的带宽消耗降低大约 50%,容量增益允许更快的查询处理,减少基础设施而不会降低性能。

1 NoSQL 存储系统

现在的 NoSQL 存储系统按存储方式的不同分为 4 大类,接下来讨论 4 类 NoSQL 系统的优缺点。

(1)键值(key-value)存储数据库。

这一类数据库主要会使用到一个哈希表。表中有一个特定的键和一个指针指向特定的数据。key/value 模型对于 IT 系统来说,优势在于简单和易部署。但是如果 DBA 只对部分值进行查询或更新时,key/value 就显得效率低下了。

(2)列存储数据库。

列存储数据库通常是用来应对分布式存储的海量数据。列存储数据库主要优势是查找速度快、查询效率高、可扩展性强,缺点就是功能相对局限,如没有二级索引等。

(3)图形(Graph)数据库。

图形数据库优点就是使用灵活的图形模型,并且能够扩展到多个服务器上。但是图形数据库目前还无法划分子图,即对图的大小有所限制,而且图中的边无法从一个顶点回指其本身。

(4)文档型数据库。

文档型数据库比键值数据库的查询效率更高,最大的优势就是对数据结构没有严格的要求,但是缺点就是在处理动态文档集时不支持多键作为主索引,也不支持事务操作。

现在很多研究是基于文档数据库的,如上海交通大学陈敏敏的《基于 MongoDB 云存储平台的论坛信息抽取与存储研究》[4] 和北京交通大学刘一梦的《基于 MongoDB 云数据管理技术的研究与应用》[5],都从理论和实践中证明了文档数据库 MongoDB 的优越性。Zachary Parker 等人[6] 也对 MongoDB 和 SQL 数据库做了性能测试的比较,包括插入、更新和选择等方面。文中就文档型数据库在处理动态文档集时不支持多键作为主索引的缺点展开研究,提出了一种组合索引技术。

2 查询机制

2.1 传统的查询方法

(1)分而治之。

这种方法忽略了底层的键值映射,初始节点发送一个查询到网络中的所有节点。然后,每个节点上都查看一下此查询是不是本地存储的数据,并把结果发送回发起节点。最后,此发起节点结合了所有的部分结果,产生最终的结果。常用的实施分而治之的方法是用 Map/Reduce 框架,如列存储的 BigTable 和 HBase,文档存储的 CouchDB 和 MongoDB 等。

(2)二级索引。

在 key-value 数据库中数据的 key 是数据的检索入口。为了实现对值的查询,需要对值建立一个有效的索引,称为列值索引或二级索引或辅助索引[7]。有两种不同的二级索引。最直接的是容易实现和维护的倒排索引或倒置列表,在信息检索中很流行。键值来自于索引的属性值。存储的值是一个具有特定属性值的数据对象的主键列表。第二类辅助索引是更复杂的数据结构,如分布式的 B-Trees[8]。将键-值对存储在系统中,形成最终指向数据对象的逻辑树。NoSQL 已经认识二级索引的重要性,实现越来越多的本地支持,如 BigTable,Cassandra 和 MongoDB 等。近期主流的基于二级索引的技术主要有 ITHBase(Indexed Transactional HBase)、IHBase(Indexed HBase)[9] 及 CCIndex[10]。

2.2 传统查询方法存在的不足

基于关键词的搜索通常包含不止一个条件的查询。不同的网上平台分析查询日志的研究揭示了一个查询平均有两到四个查询条件[11-12]。

对于分而治之的方法,只是本地每个节点计算查询负载增加了,如 Map/Reduce。因为在这种方法中,系统通常接触很多节点,由于在每个查询中每个节点都参与,资源和带宽方面引起的开销是很高的。

对于使用二级索引查询,若有 k 个查询索引,则每个查询又会有 k 个子查询。所有的 k 个部分结果相结合得到最终的查询结果。大多数网上平台通过逻辑与(AND)来组合搜索。因此,对于最后的结果,系统必须计算部分结果的交叉点,或通过逻辑或(OR)组合搜索要联合的结果。

无论哪种方式,部分结果必须通过网络转换,这可能会导致消耗高带宽。在更新的情况下,索引会导致额外的存储开销以及维护成本。Patrick Reynolds 和 Amin Vahda 应用布隆过滤器的技术手段来计算部分结果的交叉点,减少转移部分结果时的绝对数据大小[13],可以在一定程度上减少这些额外的消耗。虽然传统的查询方法得到了比较广泛的应用,但是研究表明,单一词汇的部分结果在合并时不能很好地扩展在分布式系统中[12]。

文中设计的组合倒排索引技术可以很好地扩展在分布式系统中。

2.3　组合索引技术

基于关键字的搜索不涉及对数据对象的非主键属性范围查询,因此对一个倒排列表建立二级索引是足够的。此外,为了减少倒排列表中键的组合查询的交叉开销,文中提出一种组合倒排索引技术,存储条件或标记的组合作为索引的键。对于文中使用的书签系统中组合倒排索引如图 1 所示。

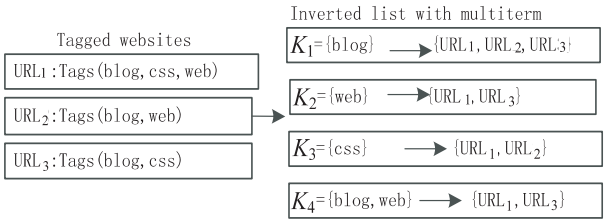

图 1　组合倒排索引

存储条件与标签的组合作为键,其中左侧是标签数据,右侧是所有可能的组合键。当单独索引的数量达到一定的阈值,会触发合并过程,将所有单独索引合并为一个整体,即一个段索引。当段索引的数量达到一定阈值时,也会触发合并操作,把若干个段索引合并到主索引中。索引合并的阈值的设置要恰当。如果阈值太小,会造成合并操作被频繁触发,系统资源消耗很大。如果阈值太大,合并操作触发的周期会很长,文档的更新不能及时反映到倒排索引文件中去,影响系统的时效性[14]。所以文中限制了条件组合的最大值,由 $S_{\max}$ 表示。让 T_p 作为页面 p 标签的数目。对于 p 来讲标签组合的数目为 $O(T_{p\max}^{S})$。大多数组合只有一次或很少被查询,而只有少数组合经常被查询。因此,文中设计出一种查询驱动优化技术,仅存储频繁访问组合。

2.4　查询处理

查询处理算法流程如图 2 所示。

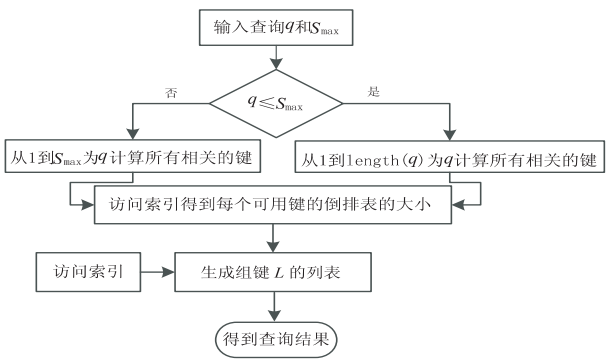

图 2　查询处理算法流程

文中采用启发式确定键访问索引的集合和顺序,对于图 1 的例子,计算访问二级索引列表的组键 L,从具有最短倒排表的键开始,迭代添加键到 L,L 最大化覆盖 q,直到涵盖所有 q 的查询条件。如果某些键实现最大化覆盖了,则节点选择这些键用作最短的倒排表。一旦发起节点计算了列表 L,就可以访问索引来检索查询结果。具体为:首先 L 被发送到所有节点,然后每一个节点将对应键的倒排表加入到中间结果中,最后一个节点将最终结果发回给发起节点。这个基本机制对于多个和单个条件查询是一致的。

2.5　索引维护

倒排索引的维护包括两个主要任务:挂起(删除)和恢复(添加)键,以及处理标签数据更新。

(1)暂停和恢复键。

为每一个键 k 提供一个长度为 l 的位向量 B_k。每一次 k 被请求,就把 B_k 右移一位,并追加 1;如果一段时间 k 没有被请求,则将 B_k 右移一位,并追加 0。B_k 所组成的数字的值代表着键 k 受欢迎的程度。如果 B_k 下降到小于系统的门限值 b^{susp} 时就挂起 k,即要将 k 的倒排表从索引中删除。如果 B_k 所组成的数字的值超过了 b^{res}(其中 $b^{\text{res}}>b^{\text{susp}}$),则恢复 k。恢复键 k 包括检索对应的倒排表,它可以对 k 执行查询,并保存 k 倒排表的结果。恢复键增加了处理用户查询的工作量,但是可以通过参数设置使得恢复键比计算用户查询更加少见。

(2)处理标签的更新。

虽然牺牲了强大的数据一致性和高可用性,但是为了限制更新时对带宽的消耗,文中提出一种通过增量更新的机制,定期更新可用的多组合键,并计算当前多条组合键的倒排表中添加或删除列表项的集合。对于单键倒排表中的每个条目,当它被添加或标记为除去时,要为其指定一个时间戳。同样地,指定时间戳记到每个组合键,用来显示最后更新时间。因此,对于组合键 K_m,可以找出所有相应的倒排表中单键的最新变化,即所有条目已添加或删除上次更新后的 K_m。增量更新相继采用了倒排表的最新变化,所有的 K_m 相应的单键更新 K_m。图 3 给出了一个例子(设 $K_1=\{\text{blog}\}$,$K_2=\{\text{css}\}$,$K_3=\{\text{blog,css}\}$)。

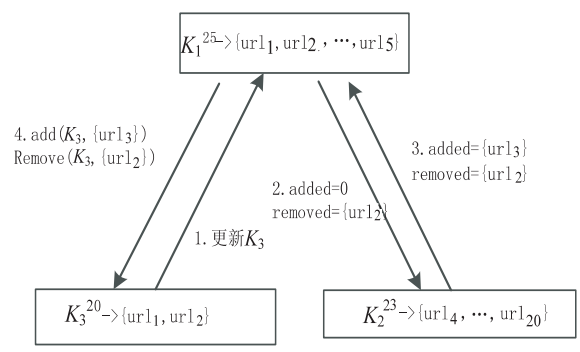

图 3　时间 $t=30$ 时键增量更新的示例

在通过组合键更新后的最大时间间隔后,可以从一个单键的倒排表中删除标记为删除的条目。在每个倒排表条目的标记代表了条目何时在倒排表中添加或删除。一个键的时间戳表示它最新的更新时间,交叉

项被标记为删除,更新后,最终 $K_3^{30} \rightarrow \{URL_1, URL_3\}$。

3 实验结果

(1)使用的数据集。

文中采用 2006 年社会书签网站 Delicious 提供的标记信息作为存储数据,其中平均每页有 4.2 个书签,平均每分钟有 67 个用户进行操作。隐私问题使得日志查询变得困难。因此,文中使用在 2006 年 3 月到 5 月之间收集的 The AOL Query Log,其中包含了约 28.8 万次查询。

(2)假设和评估方法。

文中假设用一个分布式存储系统管理倒排索引(忽略节点故障),此外设单键总是可用的。不考虑本地保存数据的布局策略,假设最坏的情况是:系统中节点数目足够多,则进行预处理查询或传播更新的相关键可以驻留在不同的节点上。实验结果以带宽消耗作为评估标准。

(3)实验设置和结果。

文中模拟每分钟 150 个用户的操作,经常更新组合键来保证组合方法和单键方法的查询结果平均重叠率达到 99%。5 次模拟了键的位向量周期转移之间的修改时间,因为它本质上影响索引的大小。

通过实验可以看出,即使延长位向量转移之间的周期,需要来存储组合键额外的存储空间也是相当低的。因此只需适当增加所需的存储,就能显著地减少带宽的消耗。图 4 显示了有更新和无更新时的带宽消耗情况。

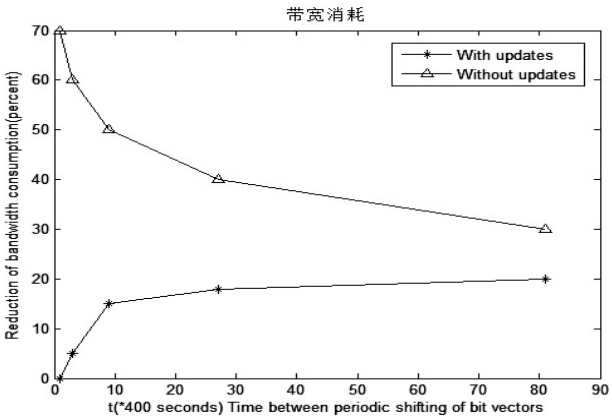

图 4 有更新和无更新时带宽消耗百分比

随着位向量定期转移的时间的增加,存储组合键的数目呈增加趋势。反过来,在执行查询时具有更好的性能,但在更新处理时需要更大的开销。结果表明,索引的查询处理性能和更新时维持索引的负载之间得到权衡。虽然提出的组合索引方法增加了索引维护的成本,但是改善整体带宽消耗的性能大大超过了维护成本的代价。尽管参数设置是假设在最坏情况下,但文中提出的方法跟单(索引)条件方法相比减少了约 50% 的带宽。

4 结束语

文中针对 NoSQL 系统支持复杂查询能力的不足,设计了一种组合索引技术。通过改进的组合条件存储的倒列表,查询驱动机制可以从最近的查询记录中自适应地存储比较受欢迎的条件组合,优化了算法,减少了相应的存储开销需求,从而降低了整体的带宽消耗,较好地改善了 NoSQL 系统的容量和响应时间。

参考文献:

[1] 陈 明. NoSQL 数据库系统[J]. 计算机教育,2013(11):107-111.

[2] 邓 攀,刘功申. 一种高效的倒排索引存储结构[J]. 计算机工程与应用,2008,44(31):149-152.

[3] 潘隆禧,孙 乐. 基于动态文档集的索引技术[J]. 计算机应用研究,2009,26(1):15-18.

[4] 陈敏敏. 基于 MongoDB 云存储平台的论坛信息抽取与存储研究[D]. 上海:上海交通大学,2012.

[5] 刘一梦. 基于 MongoDB 云数据管理技术的研究与应用[D]. 北京:北京交通大学,2012.

[6] Parker Z,Poe S,Vrbsky S V. Comparing NoSQL MongoDB to an SQL DB[C]//Proceedings of the 51st ACM southeast conference. [s. l.]:[s. n.],2013:1-6.

[7] 申德荣,于 戈,王习特,等. 支持大数据管理的 NoSQL 系统研究综述[J]. 软件学报,2013,24(8):1786-1803.

[8] Aguilera M K,Golab W M,Shah M A. A practical scalable distributed B-tree[J]. Proceedings of VLDB Endowment,2008,1(1):598-609.

[9] Kulbak Y, Washusen D. Ihbase[EB/OL]. 2010. http://github.com/ykulbak/ihbase.

[10] Zou Yongqiang,Liu Jia,Wang Shicai,et al. Ccindex:a complemental clustering index on distributed ordered tables for multi-dimensional range queries[C]//Proc of NPC'10. [s. l.]:[s. n.],2010:247-261.

[11] Chen Hanhua,Yan Jun,Jin Hai,et al. TSS:efficient term set search in large peer-to-peer textual collections[J]. IEEE Transactions on Computers,2010,59(7):969-980.

[12] Li Jinyang,Loo B T,Hellerstein J M,et al. On the feasibility of peer-to-peer web indexing and search[C]//Proceedings of international workshop on peer-to-peer systems. Berkeley:[s. n.],2003:207-215.

[13] Reynolds P,Vahdat A. Efficient peer-to-peer keyword searching[C]//Proceedings of middleware. [s. l.]:[s. n.],2003:21-40.

[14] 代万能. 倒排索引技术在 Hadoop 平台上的研究与实现[D]. 成都:电子科技大学,2013.

基于NoSQL系统的组合索引技术研究

作者： 宗平， 吴秀娟， ZONG Ping， WU Xiu-juan
作者单位： 宗平,ZONG Ping(南京邮电大学 海外教育学院,江苏 南京,210023)， 吴秀娟,WU Xiu-juan(南京邮电大学 计算机学院,江苏 南京,210023)

刊名： 计算机技术与发展 ISTIC
英文刊名： Computer Technology and Development
年，卷(期)： 2014(12)