南開大學 Nankai University

分布式数据库相似查询索引技术研究

Research on Similarity Query Index Technology in Distributed Database

答辩人：赵书楠
学号：1813055
专业：计算机科学与技术
指导教师：温延龙

南开大学　计算机学院

2022 年 5 月

目录

研究背景及意义

- 很多情况下 DBMS 的精确查询无法满足业务需求，如查询编辑错误、同一对象存在多种表示，或者检测文章抄袭、垃圾邮件等场景，因此需要引入相似查询得到相似的记录对。
- 海量数据常被存储在分布式数据库中，亟需基于分布式集群的相似查询策略。
- 本文基于前缀过滤和片段过滤提出了两种简便的相似查询倒排索引构建算法，并应用于分布式数据库，相比传统方法有效地提升了相似查询的效率。

目录

相似查询定义

- 设 r, s 为记录，R 为记录集合，τ 为相似度阈值，f 为相似度函数。
- 相似选择

$$\{r \in R | f(r, s) \geq \tau\} \tag{1}$$

- 相似连接

$$\{(r, s \in R) | f(r, s) \geq \tau\} \tag{2}$$

- Jaccard 相似度

$$Jac(r, s) = \frac{|r \cap s|}{|r \cup s|} \tag{3}$$

基于前缀的索引设计

定理 1

假设记录 x 和 y 相似，那么 x 中与 y 不匹配的元素个数 $|x| - |x \cap y| \leq |x| - \tau|x \cup y| \leq |x| - \tau|x|$。

- 如果选取 x 的任意 $|x| - \lceil \tau|x| \rceil + 1$ 个元素，其中必然存在一个元素也在 y 中出现。
- 先对记录元素按字典序升序排列，然后选取每条记录的前 $|x| - \lceil \tau|x| \rceil + 1$ 个元素，每个元素是一个前缀标签，构建倒排索引。

基于前缀的索引设计

定理 2 (前缀过滤)

如果两条记录相似，则它们内部元素排序后，前缀必然有元素重叠。

证明.

设排序后的 $x = \{x_1, ..., x_i, ..., x_{|x|-\lceil\tau|x|\rceil+1}, x_{|x|-\lceil\tau|x|\rceil+2}, ..., x_l, ..., x_{|x|}\}$，$y = \{y_1, ..., y_k, ..., y_{|y|-\lceil\tau|y|\rceil+1}, y_{|y|-\lceil\tau|y|\rceil+2}, ..., y_j, ..., y_{|y|}\}$，假设前缀没有重叠，必然存在 $x_i = y_j, x_l = y_k$，这与元素有序矛盾。 □

前缀索引构建方法

- 对数据库中每条记录进行清洗，元素按照字典序升序排列。设定 $\tau = 0.6$，获取前缀标签，并抛弃停用词。
- “[research] on [similarity] [query] index technology in distributed database”
- 相同标签所属的记录通过 groupByKey 操作进行合并，索引文件使用哈希分区划分成多个小型索引文件存储在 HDFS 上，即 $Hash(Sig) \equiv i(\mod n_p)$。

基于片段的索引设计

定理 3

假设记录 x 和 y 相似，那么 x 与 y 不匹配的元素个数 $|x \cup y| - |x \cap y| \leq |x \cup y|(1-\tau) \leq \frac{|x \cap y|}{\tau}(1-\tau) \leq \frac{1-\tau}{\tau}|x|$。

- 如果将记录 x 和 y 均划分成 $\lfloor \frac{1-\tau}{\tau}|x| \rfloor + 1$ 段，则它们必然至少存在一个完全匹配的片段。
- 为了构建索引，将每条记录依据其长度 $|x|$ 平均划分为 $\lfloor \frac{1-\tau}{\tau}|x| \rfloor + 1$ 段作为片段标签，构建倒排索引。
- 平均划分：令 $k = |x| - \lfloor \frac{|x|}{\eta_{|x|}} \rfloor \cdot \eta_{|x|}$，则 x 划分后前 k 个片段长度为 $\lceil \frac{|x|}{\eta_{|x|}} \rceil$，其余片段长度为 $\lfloor \frac{|x|}{\eta_{|x|}} \rfloor$，能够保证所有片段长度之差至多为 1。

基于片段的索引设计

定理 4 (片段过滤)

当 $\tau = 0.6$ 时，如果两条记录相似，则它们各自平均划分后至少存在一个完全匹配的片段。

证明.

由于 $\frac{|x \cap y|}{|x \cup y|} = \frac{|x \cap y|}{|x|+|y|-|x \cap y|} \geq 0.6$，所以 $|x| + |y| \leq \frac{8}{3}|x \cap y|$。又由于记录 x 中 $\frac{1}{3}|x|$ 个元素单独作为一个片段，另外 $\frac{2}{3}|x|$ 个元素相邻两两组合作为长度为 2 的片段，假设记录 x 和 y 所有片段均不匹配，那么它们不匹配元素个数必然大于等于单个元素的片段数之和，即 $|x \cup y| - |x \cap y| \geq \frac{|x|}{3} + \frac{|y|}{3}$，从而 $|x| + |y| \geq 3|x \cap y|$，与 $|x| + |y| \leq \frac{8}{3}|x \cap y|$ 矛盾。 □

片段索引构建方法

- 对数据库中每条记录进行清洗。设定 $\tau = 0.6$，使用平均划分法划分每条记录得到索引端片段标签，并抛弃停用词。
- “[research on] [similarity query] [index technology] in [distributed] [database]”
- 相同片段标签所属的记录通过 groupByKey 操作合并成记录列表，索引文件使用哈希分区划分成多个小型索引文件存储在 HDFS 上。

相似查询过程

- 相似选择：查询 s 获取前缀标签/片段标签，哈希后找到相应索引文件，内部找到相应索引项，计算相似度来验证相似。
- 相似连接：索引文件内部每个标签对应的倒排列表中的记录两两计算相似度来验证相似。
- 设分区数为 p_n，则相似选择的时间复杂度为 $O(\frac{n}{p_n})$，相似连接的时间复杂度为 $O(n)$。

目录

系统框架

- 大数据分布式处理框架 Hadoop 包含了用于分布式存储的核心组件 Hadoop 分布式文件系统 HDFS 和用于分布式计算的 Hadoop MapReduce 计算模型。
- 分布式内存计算框架 Spark 利用特有的数据抽象 RDD，将一系列 Map 和 Reduce 的操作构成一个 Spark 任务，使得整个任务都在内存中进行。

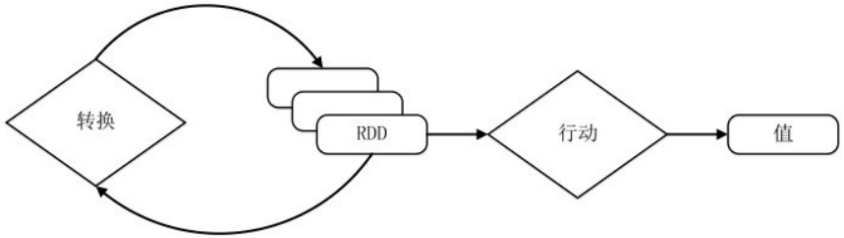


图 1: Spark 工作流程

系统框架

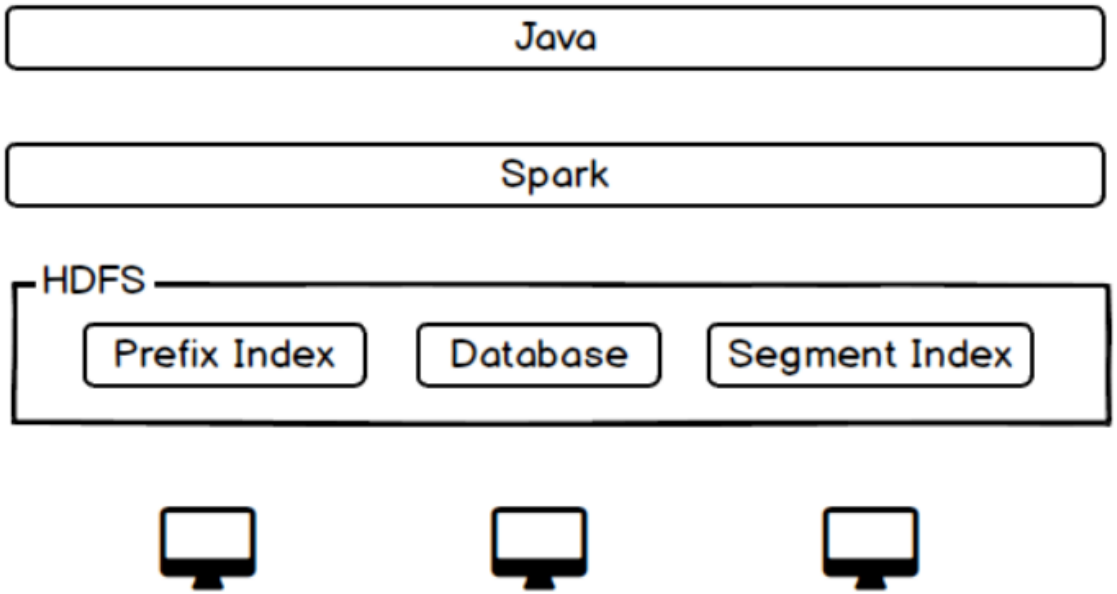
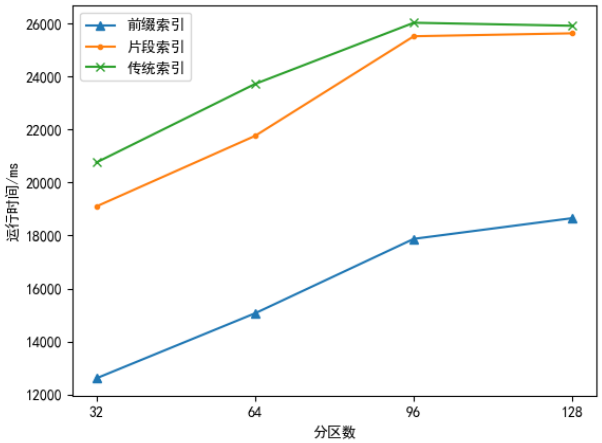


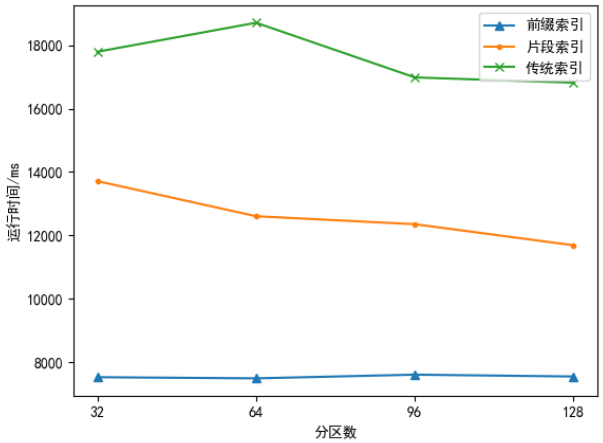
图 2: 分布式数据库相似查询系统框架

实验设置

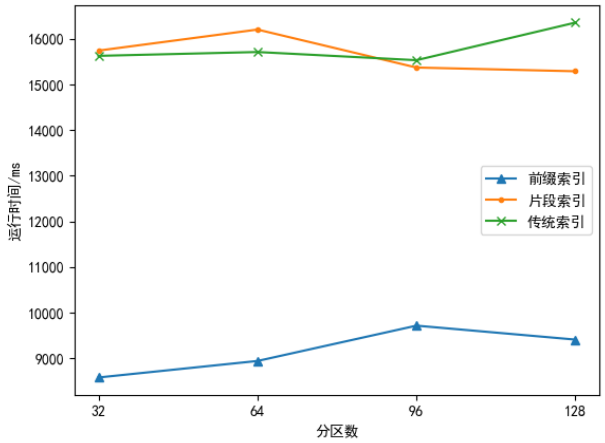
- 数据集：截止日期为 2015 年 3 月 2 日的 DBLP 数据集，是一个计算机科学领域的英文文献元数据数据集。总大小为 71.9MB，共包含 1000000 条记录，其中最小长度为 4 个单词，最大长度为 110 个单词。
- 实验环境：一个三节点的分布式集群，Ubuntu18.04.6 操作系统，AMD Ryzen 5 4500U@2.30GHz 六核 CPU，16GB 内存。



(a) 索引构建时间对比



(b) 相似选择时间对比



(c) 相似连接时间对比

实验结果分析

- 索引构建：前缀索引平均比传统索引节省了 33.4% 的时间，片段索引节省了 4.6% 的时间。
- 相似选择：前缀索引平均比传统索引节省了 57.1% 的时间，片段索引平均节省了 28.4% 的时间。
- 相似连接：前缀索引平均比传统索引节省了 42.0% 的时间，片段索引节省了 6.7% 的时间。

目录

总结与展望

总结

1. 本文基于前缀过滤和片段过滤提出了两种适用于分布式数据库的简便的相似查询索引构建方法，并给出了数学证明。
2. 经实验验证，两种索引构建方法相比传统索引构建方法均提升了相似查询效率，其中前缀索引的效率最高。

展望

未来可以研究片段索引标签选择技术，增加节点和 CPU 内核数探究合适的分区数量，设计研究支持事务型数据库的索引。

谢谢观看！