基于列存储的大数据采样查询处理

齐 文[1] 鲍玉斌[2] 宋 杰[3]

(辽东学院信息工程学院 辽宁 丹东 118000)[1] (东北大学计算机科学与工程学院 沈阳 110819)[2]

(东北大学软件学院 沈阳 110819)[3]

摘 要 大数据时代的到来给传统的数据查询带来了性能挑战,即使查询算法有着 O(n)的线性复杂度,但当 n 极大时其时间开销也难以满足用户需求。在很多实际应用中,人们并不需要精确的查询结果,但要求在给定时间内完成查询,因此可适当牺牲查询精度以满足性能约束。采样查询通过约简查询范围来提高查询性能,现有的采样方法多针对特定的算法和特定的应用场景,缺乏大数据环境下一般性的采样查询方法以及保证性能和精度的研究。文中研究大数据环境下列存储的采样查询处理,从数据划分和数据采样两方面改进大数据的查询效率。提出了基于加速比和势分布的采样方法,其支持各类采样算法,实现了分布式环境下采样查询的随机性保证、性能保证和近似性评价,并兼容了精确查询。该方法可以快速应用到已有大量数据的列存储中,具备良好的扩展性和可维护性。以 Top-K 为查询用例的实验结果证明,在不同数据量、不同数据分布和不同采样算法下,实际采样率与给定采样率的误差低于 2%,查询准确度(Accuracy)稳定,方差在 0.10 和 0.12 之间,因此提出的基于段势的数据划分的采样效率高于平均划分和线性划分。

关键词 大数据,列存储,采样查询,数据划分,加速比

中图法分类号 TP391 **文献标识码** A **DOI** 10.11896/jsjkx.190500155

Column-oriented Store Based Sampling Query Process on Big Data

QI Wen[1] BAO Yu-bin[2] SONG Jie[3]

(School of Information and Engineering, Eastern Liaoning University, Dandong, Liaoning 118000, China)[1]

(School of Computer Science and Engineering, Northeastern University, Shenyang 110819, China)[2]

(Software College, Northeastern University, Shenyang 110819, China)[3]

Abstract The era of big data bring performance challenges to traditional data query, even if the query algorithm is O(n) linear complexity, but when the n is extremely large, its time cost is also unbearable. In many practical applications, exact query results may be unnecessary but the queries should be accomplished at a given time, so appropriately losing the query accuracy is acceptable to meet performance constraints. Sampling queries can improve query performance by reducing query ranges. Existing researches are often studied for specific algorithms and specific application scenarios, and there is a lack of research on general sampling and query methods in the big data environment, as well as research on performance and accuracy guarantee. This paper studied the sampling and query processing in the big data environment, which improves the query efficiency of big data from data partition and data reduction. This paper proposed a sampling method based on speedup and potential distribution, which supports all kinds of sampling algorithms, and achieves randomicity guarantee, performance assurance and approximation evaluation of sampling queries in distributed environment, and is compatible with precise queries. This method can be applied to the column store for the big data with good expansibility and maintainability. The experimental results show that as the Top-K query case, the proposed method has better loading performance, while the sampling errors are less than 2%, and the variances of query accuracy are between 0.1 and 0.12 under various sampling rates, data volumes and sampling algorithms. The sampling efficiency of proposed partition is also higher than that of linear partition based or uniform partition based sampling.

Keywords Big data, Column-oriented store, Sampling query, Data partitioning, Accumulation ratio

---

到稿日期:2019-05-28 返修日期:2019-07-22 本文受国家自然科学基金(61672143,61433008)资助。

齐 文(1974-),男,硕士,副教授,CCF 会员,主要研究方向为大数据查询与分析,E-mail:ddqiwen@163.com(通信作者);鲍玉斌(1968-),男,博士,教授,CCF 高级会员,主要研究方向为数据仓库;宋 杰(1980-),男,博士,教授,CCF 高级会员,主要研究方向为大数据存储与管理、高能效计算。

1 引言

与传统的关系数据库以记录或者行为单位进行数据处理不同,列储存(Column-oriented Data store)以关系表中的列或列簇为单位对数据进行存储和查询等处理,便于在列上对数据进行轻量级压缩,能够降低存储成本,同时有利于迅速查询所需要的列[1]。本文研究大数据环境下的列存储的查询优化,具体思路如下:1)即使查询算法有着 O(n)的线性复杂度,但当 n 极大时其时间开销也难以满足用户需求;2)尽管一些特殊查询算法,如 Top-k,KNN,Skyline 等,自身存在优化的空间,但对于一般意义的查询算法其自身难以进行进一步优化;3)缩小搜索范围是优化查询的主要思路,传统的数据分区和索引技术可以精确地缩小搜索范围,提高查询命中率,然而这些技术都依赖数据值域的良好划分和精细的数据结构,在大数据环境中维护一个精确的数据分区和索引的代价很高;4)在很多实际应用中,人们并不需要精确的查询结果,仅需要满足一定精度要求的近似的查询结果,因此可以适当牺牲查询精度来满足性能要求,例如人们在机场通过 GPS 定位服务寻找附近的餐馆时,并不需要非常精确的数据,相反他们对响应时间的要求会更高;5)对于一般查询,查准率比查全率更为重要,例如用户能够容忍查询返回的餐馆并非匹配条件的所有餐馆,但难以接受查询结果中包含不匹配查询条件的餐馆,这一点在大数据查询中尤为明显。

针对上述问题,本文利用采样查询通过约简搜索范围来提高查询性能,从数据划分和数据约简[2]两个方面来改进大数据的查询效率,为用户提供快速、近似且精度有保证的查询处理。本文提出段是列数据按数据量的逻辑划分,首先用户提交新的查询并给定性能要求,服务器根据性能上界计算期望的采样率,并确定查询的段;查询完成后,返回查询结果和实际的查全率以及其置信度。与传统的数据分片和索引技术不同,本文仅依赖数据量划分数据,而非数据值,这种方式使划分结构更易于维护,更适用于大数据。

本文提出的采样查询处理应满足以下目标:1)通用性,即适用于列存储上的各种采样算法;2)精度保证(Guarantees),能够保证查询结果与精确结果的近似性;3)时间响应,在损失精度的情况下,提高每次采样查询的响应速度;4)扩展性,采样查询处理算法及其对应的结构可以快速扩展以适应新数据;5)适用性,采样查询处理可以适应于不同规模的集群以及不同大小的数据集。

本文第 2 节介绍采样查询的相关技术;第 3 节形式化地定义本文研究的问题;第 4 节介绍采样查询过程;第 5 节验证采样查询算法的性能、准确度和采样效率;最后总结全文并提出下一步工作。

2 相关工作

采样是近似查询的一种简单方法,采样查询的查询结果可能不准确,但仍然具有代表性,采样查询在数据库领域有着 20 多年的历史[3-4]。用户可以在数据预计算时间或查询时间内执行采样,可以对查询结果、基表或整个数据库执行采样[5]。近年来,针对采样算法自身的研究较少,如用于大数据的无参线性采样方法[6]和启发式整群采样算法[7],而针对采样算法在具体领域的应用的研究更多。文献[8]研究了数据库采样框架在推荐系统中的应用;文献[9]将采样用于查询优化,利用基于采样的迭代过程逐渐消除查询优化方案中的错误;文献[10] 研究了 Web 服务 QoS 预测过程中采样方法对预测精度的影响;文献[11]研究了采样方法在机器学习算法超参数优化中的作用。

采样方法也被频繁应用在数据查询中。文献[12]将数字图书馆中的数据抽象为图,提出了在查询相关图(Query-related Graph)上迭代地采样查询的方法;文献[13]研究了社交网络上的采样查询方法,对社交媒体数据进行时空采样,以实时创建数据可视化(热力图)。与上述研究不同,本文提出的采样查询方法并不研究采样算法自身,而研究如何从数据角度保证采样的随机性、数据分区的均衡性和查询算法的并行性。本文的创新点在于:提出了数据段和段势的概念,利用阿姆达尔定律创建数据分区模型,在分布式的环境中求解势分布函数来确定数据逻辑分区,以支持所有采样算法在列式数据库上的实现。与本文最为相关的研究是文献[14],该文通过整群抽样方法建立查询响应时间模型,可以根据采样率来调整查询时间。与之不同的是,本文提出的方法是给定采样算法,通过数据分区模型来确保列数据库中该采样方法的随机性及其与查询响应时间之间的线性关系。此外,文献[15]也提出了基于数据分块的采样方法,与文献[14]不同的是,该文提出将数据划分为多个数据块并保存到块池(Block Pool)中,随后从块池中随机选择数据块,而非从数据集中随机选择数据项,这种做法保证了随机性且加快了采样效率。文献[14]中的数据块和本文提出的段均为数据的逻辑分区,而本文分段的目的并非为了加快采样效率,而是为了确立采样率和查询效率之间的函数关系。

3 问题的定义

在大数据环境下,优化查询性能的最核心方法有两种:1)查询并行化;2)缩小搜索范围。本文采用后者,设计思路为:缩小搜索范围以提高查询效率。首先将列数据按数据量逻辑地划分为块,且划分符合特定规律;其次根据查询的性能要求选择全部或部分逻辑块;然后采用并行计算模型完成数据检索;最后评估查询结果的准确性。基于该思路,需考虑以下几个问题:1)如何确定各逻辑块的数据量比例关系,以及查询性能和选中逻辑块数量之间的关系;2)列数据按数据量划分算法需满足何种条件为宜,且这种划分方法需要具有扩展性,同时适用于历史数据和新增数据;3)如何保证选取的样本数据具有随机性。针对以上问题,本节给出关键术语的定义。

定义 1(段,Segment) 列式数据库的数据按列存储,考虑列簇 C 包含 n 个列 $c_1,c_2,\cdots,c_n$。将该数据划分到 w 个数据段 $s_1,s_2,\cdots,s_w$ 中,数据段是对数据存储文件的逻辑划分,一个数据段对应一个或多个数据文件。

在集合论中用集合的势(Cardinality)来度量有限或是无限集合的大小。本文采用术语“势”来比较不同段的数据量大小。

定义 2(段的势和势分布,Cardinality and Cardinality Distribution) 对于某列簇,其数据段 $s_k(1\leqslant k\leqslant w)$ 中现有记录数和列簇现有记录总数之间的比值称为段的势。s_k 的势为

$\|s_k\|=|s_k|/(|s_1|+|s_2|+\cdots+|s_w|)$,其中$|s_k|$表示$s_k$现有记录数(元组个数)。设函数$g(x)$是分布函数,$x\in[0,w]$,$G(x)$是$g(x)$的原函数。若将$w$个数据段按其势的增序排序并从1至$w$按自然数编号,则编号为$k$的段的势为$s_k=\int_{k-1}^{k}g(x)\mathrm{d}x$或$G(k)-G(k-1)$。由定义可知,势分布是一个单调非递减函数。

势分布是一个确定函数,不随数量的变化而变化,且由4.2节中的加速比确定。为适应采样查询,势分布不服从平均分布。段的大小不等不会导致数据在节点上不均匀的数据布局,因为段仅是数据文件的逻辑划分,而这些数据文件在节点上的布局会影响节点间数据量的均衡性,且数据布局也会影响查询的并行性。本文暂不研究数据的均衡布局算法。

定义 3(数据分段,Data Segregation)　将一个列簇的数据按势分布划分为w个段的过程称为对该列簇的数据分段。

为简化描述,本文着重考虑一个列簇,且列簇包含两个列的近似查询处理,本文方法可以推广到有多个列簇且列簇包含任意多列的列式数据库中。同时,本文的论述与证明都是基于如下的合理假设:

(1)数据库中的数据是按列簇组织和存储的,每个列簇由一个或多个列组成。

(2)列簇上的数据块(存储单元)的总数量要远大于节点数量,因此每个节点会存在多个数据块。

(3)设置段的数据量大于节点数以保证查询的并行性。

(4)对于现有数据,其数据特征和数据块特征是可知的;对于新增数据,数据装载是可控的。

4　采样查询处理

本文针对列存储数据库,通过创建一个数据分区模型,提出利用阿姆达尔定律来完成采样查询处理,通过小规模的随机样本数据描述原始数据库,进而以较小的时间代价得到查询结果,提高了查询性能。

4.1　采样查询与采样率

本文将数据库的查询处理分为两类:1)数据查询,查询的结果是数据库的数据,如查询满足给定条件的人群的年龄,查询结果"年龄"是数据库的字段;2)聚集查询,需要在查询结果上进行聚集运算,将查询结果聚集为一个或多个值,如查询数据库中年龄大于100岁的人数,需要用count()函数对查询结果进行汇总,而Top-K查询则是将查询结果汇总为K条记录。

查询结果和精确结果之间的近似性被称为准确度(Accuracy),对于数据查询可以采用查准率来度量,对于聚集查询则需要定义新的相似度度量。准确度是本文提出的近似性查询的重要评价指标,且与查询相关,本文第5节采用余弦相似度来定义Top-K查询的准确度。由于准确度需要执行两次查询(精确与近似查询)并比较查询结果才能获得,因此在查询执行前,本文采用采样率作为度量查询的近似度。

定义 4(采样率,Sampling Rate)　参与查询的数据量与数据总量的比值。定义根据需求确定的采样率为期望采样率,记为p',期望采样率是段选择算法的重要依据;而实际的查询数据量与数据总量的比值为实际采样率,记为p。理论上,$p'\approx p$,置信度是指$p'=p$的概率。

完全随机的采样方法包括简单随机采样、分层采样和整群采样,是一种仅基于数据量的高度随机的采样方法。分层采样和整群采样都是以数据块为单位进行样本选取,由于原始数据集规模庞大,数据块数量众多,因此按块采样的采样空间太大,难以保证采样的随机性。本文对数据块进行逻辑分段,按每个逻辑段进行采样,因此段内数据块数量减少。由数据段之间的关系和段势可知,该方法的随机性更好。综上所述,分层采样和整群采样都涉及段选择和段内采样,这种采样方式既可差异化每个段的采样规模,又能保证段内采样的随机性。

定义 5(段选择,Segment Selection)　段选择是指确定包含满足查询条件的数据的所有段的过程。给定查询时间上界t,段选择应满足:1)具有随机性,避免反复查询同一段数据;2)查询时间不大于t;3)最大化查全率。

给定时间上界t,查询复杂度为$f(n)$,数据总量为N。在大数据环境下,设$t\leqslant f(N)$,则t时间内可以查询的数据量与总数据量的比例如式(1)所示:

$$p'=\frac{f^{-1}(t)}{N}\tag{1}$$

以某列簇为例,设函数$g(x)$为w个段的势分布函数,随机选择某一段a,求解b:

$$\begin{cases}p'=\int_{a-1}^{b}g(x)\mathrm{d}x, & b\leqslant w\\ 1-p'=\int_{b}^{a-1}g(x)\mathrm{d}x, & b>w\end{cases}\tag{2}$$

当$b\leqslant w$时,选择的数据段为$a,a+1,\cdots,\lfloor b\rfloor$。当$b>w$时,选择的数据段为$1,2,\cdots,\lfloor b\rfloor,a,a,a+1,\cdots,w$。随机选择$a$能够保证段选择的随机性,避免势偏小的段总是被查询,而势大的段难以被查询。

根据段选择算法,不失一般性,设b为自然数且$b\leqslant w$,那么最终选择的数据块为$a,a+1,\cdots,b$。根据算法,实际的数据量为$\sum_{k=a}^{b}|s_k|$,因此实际的查全率为:

$$p=\sum_{k=a}^{b}|s_k|/\sum_{k=1}^{w}|s_k|\tag{3}$$

简单随机采样作为其他各种采样形式的基础,基于绝对的随机性,不考虑数据的分段情况,这使得每一个样本都有相同的概率被抽中,随机地从任意段内选择一定规模的数据量进行查询,对任何支持随机采样的查询方式都有良好的适应性。

本文提出的采样查询的步骤为:首先,用户提交新的查询并给定性能要求,服务器根据性能上界,计算期望的查全率,并采用段选择算法确定参与查询的数据(段);其次,查询完成后,返回查询结果,并返回实际的查全率以及其置信度,置信度取决于数据分段方法,即段势是否完全服从势分布。

4.2　加速比和势分布

本节讨论势分布函数的确定。根据定义2,w个段的势之和为1,且每个段势均在$(0,1]$区间内,因此,势分布函数应该是满足$0<\int_{k-1}^{k}g(x)\mathrm{d}x\leqslant 1$且$\int_{0}^{w}g(x)\mathrm{d}x=1$的分布函数。

由于并行性的存在,NoSQL数据库的查询性能和数据量并非成正比关系,当数据量增加时查询性能并非等比例地降低;而由上节可知,本文期望在给定时间内确定最大查询范围,并计算查全率。因此,当查询算法的复杂度为线性时(如线性搜索),查询时间t和段的编号x之间满足线性正相关$t(x)=kx$。而查全率p则与段势$g(x)$线性正相关。可以很

容易地构造常数函数 $g(x)=1/w$，表示段势服从平均分布。平均势分布函数结构简单，但存在以下弱点：减少一个段，查全率等比例降低，但查询时间与段势之间的关系复杂，难以按查询时间选择段。因此，需要设计 $g(x)$ 函数使 $t(x)=kx$。

在分布式系统中存在加速比的概念，可以通过增加节点来提高分布式程序的性能，但当节点增加时，性能的提高会逐渐减少，最终达到一个稳定的加速比。阿姆达尔定律指出，系统某一部件采用某种更快的执行方式后，整个系统功效的提高与这种执行方式的使用频率占总时间的比例有关。而由并行方法所获得的加速比为：

$$\frac{T_s}{T_p}=\theta=\frac{1}{1-z+\frac{z}{r}} \tag{4}$$

其中，z 为问题中可被并行处理的比例，$(1-z)$ 是被串行处理的比例，r 为并行处理节点的数量，θ 为并行执行用时（T_p）相比于串行计算用时（T_s）的加速比。式(4)的函数图像如图 1 所示。根据式(4)可以引出两条重要理论：

(1)当 z 很小，即程序中可并行化部分较少时，将处理过程并行化，优化效果不明显。

(2)当 r 无限大时，$\lim\limits_{r\to\infty}\frac{z}{r}=0$，则最后总加速比的极限为 $\frac{1}{1-z}$，这意味着并行运算的最终性能仍会受到程序中串行部分的限制。

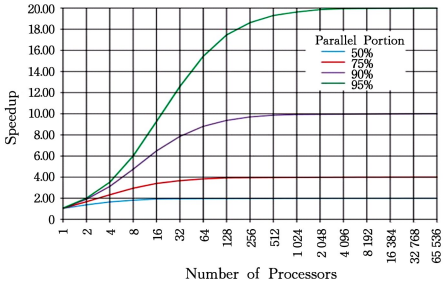


图 1 阿姆达尔定律的函数图像示例

Fig. 1 Images of Amdahl's law

设 N 为总数据量，M 为数据块大小，记 $X=\int_{x-1}^{x}g(y)\mathrm{d}y$，那么 NX 为第 x 个段的数据量，NXM^{-1} 为第 x 个段的块个数。若数据块分散性良好，则认为 NXM^{-1} 是查询第 x 个段时的最大并行程度。对于第 x 个数据块，特定计算模型（如 *MapReduce*）实现的查询算法中 z 值已知，若采用串行方法查询，查询时间与数据量成正比，设查询时间为 $aNX+b$ $(a,b>0)$，则第 x 个段的查询算法中查询执行节点数目 $r(x)$、加速比 $\theta(x)$、查询时间 $t(x)$ 的计算方法如式(5)所示：

$$r(x)=NXM^{-1}$$

$$\theta(x)=\frac{1}{1-z+\frac{z}{r(x)}}=\frac{1}{1-z+\frac{zM}{NX}} \tag{5}$$

$$t(x)=(aNX+b)\theta(x)^{-1}=(aNX+b)(1-z+\frac{zM}{NX})$$

由于 $t(x)=kx$，因此可知：

$$kx=(aNX+b)(1-z+\frac{zM}{NX}) \tag{6}$$

整理式(6)，可得式(7)是关于 X 的一元二次方程：

$$a(1-z)N^2X^2+(aNMz+bN(1-z)-kxN)X+bzM=0 \tag{7}$$

将式(7)简单记为 $AX^2+BX+C=0$，X 的两个根为 $X=[-B\pm(B^2-4AC)^{1/2}]/2A$，可知 A 和 C 都为正数，因此保留 X 为正数的解 $X=[(B^2-4AC)^{1/2}-B]/2A$。根据式(7)，B 中包含 x，A 和 C 中都不含 x。

$$\frac{1}{2A}(\sqrt{(B^2-4AC)}-B)=\int_{x-1}^{x}g(y)\mathrm{d}y \tag{8}$$

在式(8)两边对 x 求导，设左式对 x 的导数为 $\varphi(x)$。由于段编号从 1 开始，因此可以认为第 0 段的势为零，$g(0)=0$，得到 $g(x)$ 的函数表达式为：

$$\varphi(x)=g(x)-g(x-1)$$

$$g(x)=g(x-1)+\varphi(x)=\varphi(1)+\varphi(2)+\varphi(3)+\cdots+\varphi(x) \tag{9}$$

其中，M 和 N 的取值与执行环境相关，a 和 z 的取值与算法相关，k 和 b 的值为自定义值。$\varphi(x)$ 的形式如式(10)所示：

$$\varphi(x)=\frac{p_1x+p_2}{\sqrt{p_3x^2+p_4x+p_5}} \tag{10}$$

其中，p_1-p_5 均为由 a,b,N,M,z 确定的常数。

4.3 数据分段

数据分段的目标是将数据按式(10)推导的段势分布划分成 w 段，w 值根据实际数据量和数据文件的大小确定，一般地，为实现均衡的数据布局和提高查询并行性，w 的数据量应该远大于节点的数量。因此，针对某列簇在 w 和势分布已知的前提下设计数据分段算法。当数据库不存在数据时，数据分段实际上是按比例划分集合的过程，且当新数据装载时仅需“对号入座”即可；当数据库已经存在数据时，为了避免现存数据在文件块中移动，分段方法有所不同。

以某列簇为例，当数据库中该列簇已经存在数据时，我们将对所有数据文件上的记录条数进行统计，并按照一个数据文件一个段的方式进行划分，称其为文件段，按照给定的段分布将文件段合并成新段，可以通过多元方程组确定合并策略。一般地，数据库的数据文件大小是固定的，如 HBase 采用 HDFS 作为文件系统，每个数据块的大小固定为 64 MB，因此每个文件段的势近似相等。

若数据文件大小不固定，则需要计算每个段的势，按照段的势升序排序。文件段合并过程类似于背包问题，将采用贪婪法求解。由于合并后的段的数量 w 未预先指定，相当于背包问题中的包有无限多个，因此合并算法总能找到较优解。在合并过程中，若出现冲突，则尽量保证段的势的实际值略大于势分布的计算值。

5 实验评估

实验的主要目的是评估本文提出的采样查询的效率和效果，对段号和查询时间的线性关系、采样 Top-K 查询的性能和误差进行分析。

5.1 实验准备

实验在 1 个由 17 个节点构成的集群上执行，其包含 1 个管理节点和 16 个数据节点。17 个节点都是同构的清华同方超翔 Z900 计算机（Inter Core-i7-7700，3.10 GHz，8 GB，1 TB），千兆网络环境，CentOS 7 操作系统。大数据开发平台是 Hadoop，版本为 2.7.1，使用 MapReduce 编程模型。

本文提出的采样查询与数据类型、列簇中列的个数均无关,因为采样算法总是逐列完成采样查询。为了增加数据条数,设计数据集仅包含一个列簇,而每个列簇只有一个浮点数属性列,属性值域为[0,10000]。数据生成算法为该值域内的浮点随机数,且服从平均分布,可采用 Java 的 Random.nextDouble(10000)方法生成。

实验中,HDFS 复制因子为 3,数据块大小为 128 MB。实验过程中生成了最大 300 GB 的列簇数据(由于块复制,实际数据量接近 1TB),设定最小段的数据量大约为 8 GB (16 个数据块),确保每个段的数据都能分配到所有节点上,并按实验数据量计算分段规模。该设置满足第 3 节给出的条件:数据块远大于节点数量,段的数目远大于节点数目。实验最终采用的分布函数是一个近似函数,$g(x)=0.004\sqrt{x}$,是式(10)在函数图像上的合理近似。

如前文所述,本文提出的采样查询处理具有通用性、精度保证、时间响应短、扩展性好和适用性强的优势。在查询实验中,将在不同数据量、不同分区方法和不同采样算法下执行 Top-K 查询,以验证采样查询处理的通用性和适用性。最后重点对比段势分区下的采样查询和平均分区或线性分区下的采样查询,结果表明,在相同精度保证下,前者的时间响应更短。其中,段势/平均/线性分区是指各个区的数据量满足段势/平均/线性分布。实验中记录执行时间 T(Time)、查询采样率 S(Sampling)和查询准确度 A(Accuracy)。定义查询准确度 A 为采样查询结果集和常规查询结果的余弦相似度,并采用多次采样查询结果的平均值。实验度量:1)期望采样率(SE)和实际采样率(S)的误差 $E=|SE-S|/SE$,以讨论数据分区对采样精确性的影响;2)本文期望在较短时间内获得较高的准确度,而查询执行时间与数据量以及数据的并行性有关,其中数据量与采样率 S 相关,并行性与数据分区有关。事实上,采样率 S 也取决于数据分区,因此采样率可以表征查询时间。我们期望:采样较少的数据获得较高的准确度。综上,本实验度量查询准确性随采样率增加而提高的程度,也即准确度和采样率的比值,称之为采样效率,用 A/S 来表示,采样效率是一个无量纲的值。

实验记录 Top-K 采样查询在 3 种采样方法、3 种数据分区方法、5 种规模的数据集和 3 种期望采样率条件下的 T、实际采样率 S 和准确度 A,共采集了 $5\times3\times3\times3\times3=405$ 组数据。实验条件的组合如表 1 所列。

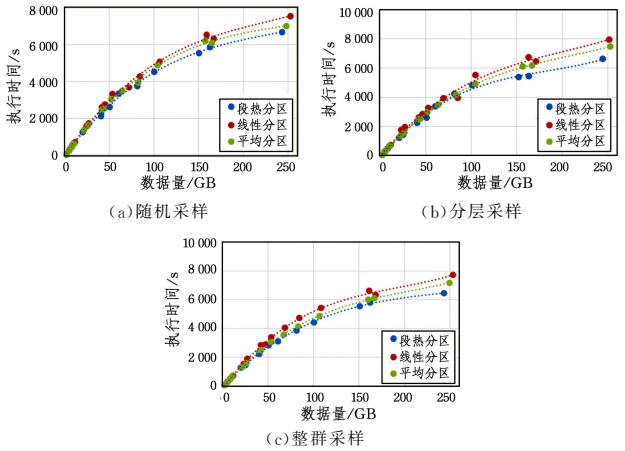
表 1　Top-K 实验案例

Table 1　Experimental cases of Top-K query

数据量/GB	采样方法	数据分区方法	期望采样率	测量值
10,50	随机采样	段势分区	0.2	执行时间 T
100,200	分层采样	线性分区	0.5	准确度 A
300	整群采样	平均分区	0.8	采样率 S

5.2　执行时间的分析

Top-K 采样查询的执行时间与数据量以及并行程度有关,其中数据量是数据集大小和实际采样率的乘积,而并行性则与数据分区有关,同一种分区方法下的并行性差别不大。通过实验分别比较了 3 种采样方法下的数据量和执行时间的关系,结果如图 2 所示。



(a)随机采样　(b)分层采样

(c)整群采样

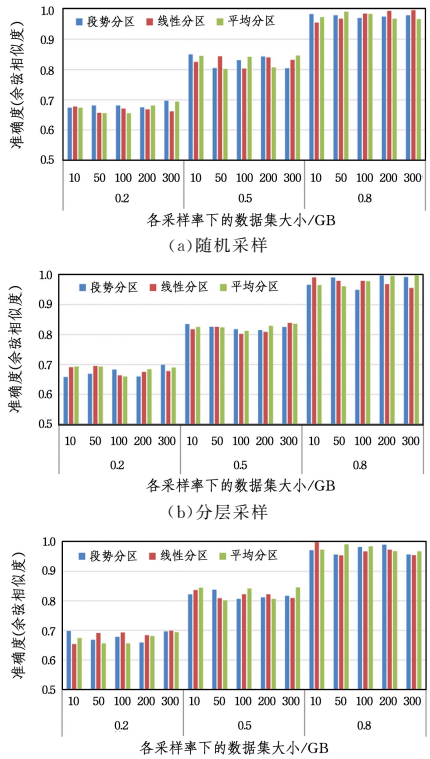
图 2　Top-K 采样查询性能

Fig. 2　Performance of Top-K sampling queries

从图 2 中的散点图和趋势线可以明显地看出,查询执行时间与数据量是强相关的,无论采用何种数据分区以及何种采样方法,查询性能均相差不大。但本文提出的基于段势的采样查询在任何采样算法下的性能均有优势,且当数据量较大时优势更为明显,这是因为段势分布使采样结果有着良好的并行性,与之对比的是线性划分,当数据量增加时,其查询性能较本文方法的差距明显。

5.3　查询准确度的分析

由于实验采用的数据量大,数据在值域上分布密集,又由于采用余弦相似度作为查询准确度的度量,较小的误差难以从准确度数值上反映出来,因此采样 Top-K 的查询准确性很好,在采样率为 0.2 时准确度大于 0.6,在采样率为 0.5 时准确度达到 0.8 以上,当采样率为 0.8 时,准确度已经接近 1。由此可见,Top-K 查询适合采用采样的方式来完成。



(a)随机采样

(b)分层采样

(c)整群采样

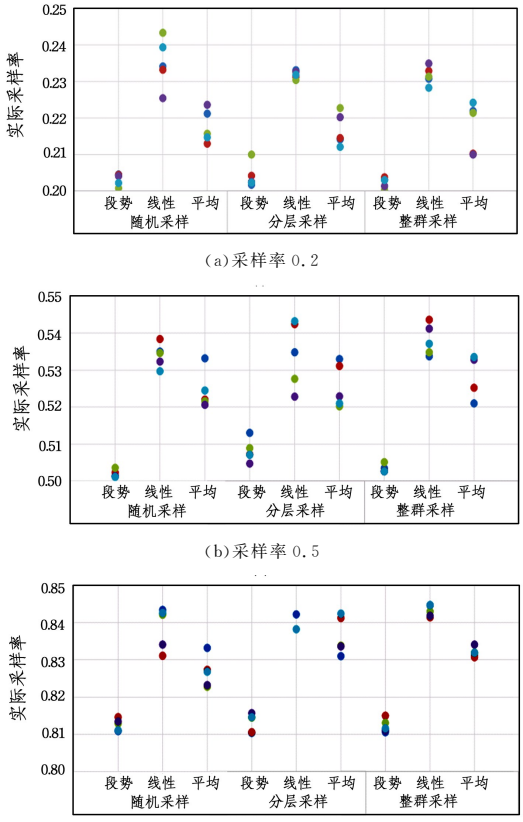
图 3　Top-K 采样查询准确度

Fig. 3　Accuracy of Top-K sampling queries

从图 3 中可以看出,准确度和采样率是紧密相关的,且与数据集大小无关,这是因为实验采用的数据在值域上分布平均。实验中,实际采样率与数据分区有着密切的关系,且与期望采样率存在偏差。由图 3 可知,段势分区采样方法较其他方法并没有明显的优势,几种方法的准确度接近并且稳定。由于采用了余弦相似度而非集合相似度作为准确性度量,因此准确性度量存在很大的随机性,后文会详细分析。段势分区在真实采样率、采样误差和采样效率上有优势,但由于各种数据分区和采样方法在采样率上存在差异,因此查询准确度也存在细微差别。

5.4 真实采样率和采样误差的分析

尽管本文提出的基于段势分区的采样方法在查询准确性上的优势不明显,但其有着非常精确的采样率,实际采样率和期望采样率的差距非常小,如图 4 所示。



(a)采样率 0.2

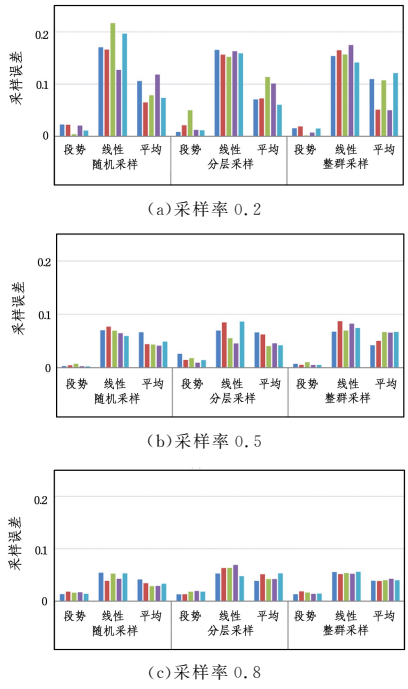
(b)采样率 0.5

(c)采样率 0.8

图 4 Top-K 采样查询实际采样率

Fig. 4 Actual sampling ratios of top-k sampling queries

图 4 中,每一个数据点表示一种数据集在不同的分区方法、不同的采样率和不同采样算法下的实际采样率,每个图中 y 轴的最小刻度为期望的采样率。由图 4 可知,段势分区的采样算法较其他分区有着最为精确的采样率。在采样过程中,很难对一个数据文件的局部进行采样,一般情况下,或选中数据文件,或不选择该文件。段势分区下数据文件大小不同且满足势分布,可以减少采样误差,而线性分区下的数据文件大小尽管各有不同,但不符合势分布,采样时存在误差。平均分区的误差最大,因为在平均分区中,每个文件的大小是一致的。图 5 进一步分析了 3 种分区方法的采样误差,清晰地反映了本文提出的方法在采样精度上的优势。



(a)采样率 0.2

(b)采样率 0.5

(c)采样率 0.8

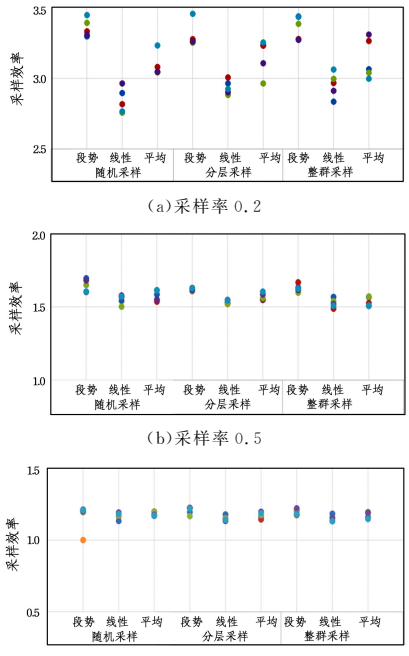
图 5 Top-K 采样查询采样误差

Fig. 5 Sampling errors of Top-K sampling queries

由图 2 和图 3 可知,本文提出的采样查询方法的查询速度快,且采样数据最少,最接近采样要求,查询准确性较其他方法并无损失。因此,所提方法的采样效率很高,下一节将重点比较采样效率。

5.5 采样效率的分析

我们期望:较短时间内获得较高的准确度。根据前文的分析,该期望等同于采样较少的数据获得较高的准确度。本实验将准确度和采样率的比值称为采样效率,用 A/S 表示,如图 6 所示。图 6 中,每个数据点表示数据集在不同数据分区、采样算法和期望采样率下的采样效率。



(a)采样率 0.2

(b)采样率 0.5

(c)采样率 0.8

图 6 Top-K 采样查询采样效率

Fig. 6 Sampling efficiency of top-k sampling queries

由图 6 可知,在较小的采样率下,本文提出的采样方法的采样效率有着明显的优势,如图 6(a)所示,段势分区下的采样效率较线性分区和平均分区平均高出 15%和 10%。但随着采样率的提升,参照前文的准确度分析,各个采样查询的准确度都很高,因此段势分区的采样效率优势逐渐降低,但采样效率仍然略高于其他两者。

5.6 实验总结

本实验涉及数据量、采样方法、分区方法、执行时间、期望采样率、准确度、实际采样率、采样效率等测量属性,这些属性之间的关系复杂,具体如表 2 所列。

表 2　实验各测量属性间的关系

Table 2　Relationships among measured attributes in experiments

	数据量	采样方法	分区方法	执行时间	期望采样率	准确度	实际采样率	采样效率
数据量	—	无关	无关	正相关	无关	无关	无关	无关
采样方法	无关	—	无关	无关	无关	无关	无关	无关
分布方法	无关	无关	—	具有优势	无关	具有优势	具有优势	具有优势
执行时间	正相关	无关	具有优势	—	正相关	正相关	正相关	无关
期望采样率	无关	无关	无关	正相关	—	正相关	正相关	无关
准确度	无关	无关	具有优势	正相关	正相关	—	正相关	无关
实际采样率	无关	无关	具有优势	正相关	正相关	正相关	—	无关
采样效率	无关	无关	具有优势	无关	无关	无关	无关	—

本文提出的段势分区较常见的线性分区和平均分区有着采样精确和查询准确的优势:实际采样率和期望采样率的误差很小,采用较小的数据集可以得到准确的查询结果。该方法查询性能的提高得益于查询并行性的提高。

结束语　本文针对大数据环境下的采样查询问题,提出对数据按给定分布进行分区,然后在查询过程中约简查询数据,实现满足给定查询响应时间的最大精度采样查询。本文基于阿姆达尔定律推导了段势分布和加速比,在分布式环境中保证了查询时间和采样率之间的线性关系。实验结果表明,本文提出的采样查询方法能够提高查询的效率并且具有良好的可扩展性和适应性。此外,在研究过程中还存在一些有待改进之处,主要是进一步细化数据分区的粒度,再通过数据约简来更好地消除信息过载带来的负面影响。

参 考 文 献

[1] SHEN D R, YU G, WANG X T, et al. Survey on NoSQL for management of big data[J]. Journal of Software, 2013, 24(8): 1786-1803. (in Chinese)
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(8):1786-1803.

[2] LUO C, JIANG Z, HOU W C, et al. A sampling approach for skyline query cardinality estimation[J]. Knowledge and Information Systems, 2012, 32(2): 281-301.

[3] GIBBON P B. Approximate Query Processing: Taming the TeraBytes[C]// International Conference on Vldb. DBLP, 2001.

[4] GRAHAM C, GAROFALAKIS M, HAAS P T, et al. Synopses for Massive Data: Samples, Histograms, Wavelets, Sketches[J]. Foundations and Trends in Databases, 2011, 4(1/2/3): 1-294.

[5] CHAUDHURI S, DING B, KANDULA S. Approximate Query Processing: No Silver Bullet[C]// the 2017 ACM International Conference. ACM, 2017.

[6] LIU L, HU G. A Parameter-Free Linear Sampling Method[J]. IEEE Access, 2019, 7: 17935-17940.

[7] ZHAO J, SUN J, ZHAI Y, et al. A Novel Clustering-Based Sampling Approach for Minimum Sample Set in Big Data Environment[J]. International Journal of Pattern Recognition and Artificial Intelligence, 2018, 32(2): 4.

[8] HAMIDI H, MOUSAVI R. Analysis and Evaluation of a Framework for Sampling Database in Recommenders[J]. Journal of Global Information Management, 2018, 26(1), 41-57.

[9] WU W, NAUGHTON J F, SINGH H. Sampling-based query reoptimization[C]// Proceedings of the 2016 International Conference on Management of Data. ACM, 2016: 1721-1736.

[10] LI J, LIN J. Research on the Influence of Sampling Methods for the Accuracy of Web Services QoS Prediction[J]. IEEE Access, 2019, 7: 39990-39999.

[11] DECASTRO-GARCÍA N, MUÑOZ CASTAÑEDA Á L, ESCUDERO GARCÍA D, et al. Effect of the Sampling of a Dataset in the Hyperparameter Optimization Phase over the Efficiency of a Machine Learning Algorithm[J]. Complexity, 2019, 2019: 1-16.

[12] LIU W, SU J. Online digital library sampling based on query related graph[J]. The Electronic Library, 2018, 36(6): 1082-1098.

[13] STOEHR N, MEYER J, MARKL V, et al. Heatflip: Temporal-Spatial Sampling for Progressive Heat Maps on Social Media Data[C]// 2018 IEEE International Conference on Big Data (Big Data). 2018: 3723-3732.

[14] ZHANG J, NIU B. A clustering-based sampling method for building query response time models[J]. Computer Systems Science and Engineering, 2017, 32(4): 319-331.

[15] HE Y, HUANG J Z, LONG H, et al. I-Sampling: A New Block-Based Sampling Method for Large-Scale Dataset[C]// EEE International Congress on Big Data (BigData Congress). 2017: 360-367.