文章编号：2095-2163(2023)12-0056-06 中图分类号：TP311.5 文献标志码：A

关系型大数据查询的层次结构

胡天伟[1]，余 靖[2]，刘国华[1]，陈德华[1]
（1 东华大学 计算机科学与技术学院，上海 201620；2 燕山大学 信息科学与工程学院，河北 秦皇岛 066004）

摘 要：数据查询是获取信息的重要手段，大数据的特征从不同维度上给经典的数据查询理论和方法带来了新的挑战，如何最大限度地获取信息是大数据应用亟待解决的问题。在大数据规模庞大、时效性强、类型多样化、准确性弱等特征中，规模庞大最为突出，围绕该特征的大数据查询解答问题是目前的研究热点。本文针对数据规模庞大这一特征，以关系型数据为对象，描述了由数据库向大数据转化的演变历程，并根据大数据的特征将大数据分成八类，给出了大数据的形式化定义，以关系型大数据库为例，讨论了查询的层次结构，并将查询分类为：一阶查询、存在性查询及不动点查询。针对不同的查询，讨论了其预处理方法并进行实验。

关键词：关系型数据；大数据；查询；不动点查询层次；预处理

Hierarchy of queries on relational big data

HU Tianwei[1], YU Jing[2], LIU Guohua[1], CHEN Dehua[1]
(1 College of Computer Science and Technology, Donghua University, Shanghai 201620, China;
2 School of Information Science and Engineering, Yanshan University, Qinhuangdao Hebei 066004, China)

Abstract: Data query is an important means of obtaining information. The characteristics of big data have brought new challenges to the classic data query theory and methods from different dimensions. How to obtain information to the maximum extent is an urgent problem to be solved in big data applications. Among the characteristics of big data, such as large scale, strong timeliness, diverse types, and weak accuracy, large scale is the most prominent feature, and the query and answering of big data around this feature is a current research hotspot. Aiming at the characteristics of huge data scales, this paper describes the evolution process from database to big data with relational data as the object, divides big data into eight categories according to the characteristics of big data, and gives a formal definition of big data. Taking the relational large database as an example, the hierarchy of the query is discussed, and the query is classified into first-order query, existence query, and fixed point query. For different queries, the preprocessing methods are discussed and experiments are carried out.

Key words: relational data; big data; query; fixpoint query hierarchy; preprocess

0 引 言

在大数据的冲击下，一些被人们奉为经典的理论已经显露出不足。多项式时间复杂度是计算理论中问题易解性(tractable)的突出标志，但当数据规模庞大时，具有线性时间复杂度的查询算法求解时间变得难以容忍[1]，多项式时间复杂度这一标志的适用性遭到质疑。为了迎接挑战，李建中[2]分析了大数据应用遇到的主要障碍，归纳出攻克这些障碍必须解决的十大关键问题，大数据计算的复杂性问题位列其中。而查询解答(Query Answering)是获取满足要求信息的计算过程，在大数据环境下，这个计算过程面临新的问题，这些问题成为大数据计算复杂性研究的重要内容。

大数据查询的对象具有数据规模庞大(Volume)、时效性强(Velocity)、类型多样化(Variety)、准确性弱(Veracity)等特征，这些特征直接影响查询解答的复杂性。因此，以特征为主线是研究大数据查询解答问题的必要途径。文献[1]以数据规模庞大、时效性弱、类型单一(关系型)、准确

基金项目：国家自然科学基金(61872311)。

作者简介：胡天伟(1997-)，男，硕士研究生，主要研究方向：知识图谱、大数据查询；余 靖(1976-)，女，博士，副教授，主要研究方向：数据库理论、大数据查询、工业大数据等；刘国华(1966-)，男，博士，教授，主要研究方向：大数据查询、工业大数据、医疗大数据等；陈德华(1976-)，男，博士，教授，主要研究方向：智慧医疗、数据科学、深度学习可解释性等。

收稿日期：2022-11-29

性强为主线，揭示出一阶查询（First-Order Query）的查询解答所面临的问题，证明了这条研究路线的可行性。在这条路线上的大数据，本质上就是规模庞大的关系型数据（即关系型大数据）。文献[3]将关系数据查询进行分类，包括底层的一阶查询和上层的不动点查询（Fixpoint Query）等。

本文在文献[1]基础上，针对不动点查询层次（Fixpoint Query Hierarchy）[3]中的每一类查询，分别研究其在大数据环境下查询解答遇到的新问题。主要贡献如下：

（1）描述出了数据库向大数据的演变过程，并归纳出八类大数据，给出关系型大数据的形式化定义；

（2）明确关系型大数据上查询的种类；

（3）讨论不同种类大数据查询的预处理方法，并在大数据集上进行查询解答实验。

1　相关工作

目前，大数据查询问题的研究已形成“一个中心、两条分支”的格局。“一个中心”就是以探讨大数据查询的易解性问题为中心，“两条分支”分别是研究易解查询的界定方法，以及研究通过预处理手段使查询变得易解的方法。其中易解查询的界定方法，是研究大数据查询解答问题重要的先决条件。

计算模型是大数据理论研究的基础，Map-Reduce 计算模型是迄今为止认可度最高的大数据编程范式，围绕这一编程范式人们进行了一系列研究和探索。研究内容主要集中于如下问题：

（1）Map-Reduce 的通信复杂性问题[4]；

（2）数据库一些经典查询（如：“Join 查询”、“Skyline 查询”等）在 Map-Reduce 下的计算复杂性问题[5]；

（3）Map-Reduce 下不同规模输入的分配以及时间、空间资源的管理问题[6]；

（4）Map-Reduce 的计算边际以及 Map-Reduce 下分析查询和分布式负载均衡问题[7]；

（5）在 Map-Reduce 下消除查询中的冗余 I/O 和大数据查询优化问题[8-9]。

相关研究虽然取得了一些成果，但是 Map-Reduce 在时间效率、编程模式、I/O 复杂性和通信复杂性等方面还有待于提高和完善，目前尚不能作为进行系统理论研究的计算模型。为了摆脱这一困境，人们进行了超越 Map-Reduce 的大数据计算复杂性理论研究工作，为大数据理论研究探索出了一条新路[10]。

基于图灵机模型，文献[1]对关系型大数据易解查询界定方法进行了研究，提出了大数据查询易解性界定标准与其复杂性类别的定义。文献[11]通过实验方式，得出亚线性时间可能是大数据计算问题易解性判定标准的猜测，这一猜测得到了国内外学者的认可，文献[12]基于图灵机模型对这一猜测的正确性进行了理论证明。

对于通过预处理手段使查询变得易解的方法研究方面，文献[10]对如何通过小数据找到精确的查询结果，查询的近似求解等问题进行了讨论；文献[13]研究了通过访问有限数量的数据来查询大数据的可行性，使用视图查询有限数量的大数据，对大数据进行算法采样，从数据量大的特征入手，研究大数据查询的预处理手段；文献[14] 从大数据真实性弱这一特征出发，研究了大数据查询方法，重点研究了数据预处理阶段提高数据质量的方法，提出了对不完全数据的近似查询求解方法。

针对大数据的计算模型还在发展和完善过程中，尚不足以支撑系统的理论研究，未来的研究工作还将以图灵机为主要计算模型。大数据查询问题的研究对象目前集中于具有规模庞大这一特征的关系型大数据，但除了位于一阶查询层的查询外，其他查询在大数据环境下还有待于深入研究。

2　关系型大数据

2.1　大数据的定义

引入关系型大数据定义之前，先根据规模庞大、时效性强、类型多样化、准确性弱等特征，给出大数据的形式化定义如下：

定义 1　大数据 D_B 是一个形如 $(D,T,type,s,v)$ 的 5 元组，其中：

（1）D 是一个数据集，$|D|$ 至少是 PB 级别，$d_i \in D(1 \leqslant i \leqslant n)$ 称为数据对象；

（2）$T=\{d_1,\cdots,d_n\}$ 是一个数据类型集合 $t_i \in T(1 \leqslant i \leqslant m)$ 称为数据对象的类型，$m \leqslant n$；

（3）$type:D \rightarrow T$ 是一个类型函数，对于数据对象 $d_i \in D(1 \leqslant i \leqslant n)$，$type(d_i)=t_j$，$t_j$ 称为 d_i 的类型，$1 \leqslant t_j \leqslant m$；

（4）$s:D \rightarrow R \times R$ 是一个时间间隔函数，R 为非负的实数集合，对于数据对象 $d_i \in D$，$s(d_i)=(x,y)$ 表示 d_i 的有效时间，$1 \leqslant i \leqslant n$、（此处参照 $Unix$ 时间，x 代表生效时间，y 代表失效时间，例如用 $s(d_i)=(-631180800,1640966400)$ 可以表示此数据对象

的有效时间从 1950 年 1 月 1 日 0 时 0 分 0 秒至 2022 年 1 月 1 日 0 时 0 分 0 秒);

(5) $v:D\rightarrow[0,1]$ 是一个真实度函数,对于数据对象 $d_i \in D(1 \leqslant i \leqslant n)$, $v(d_i)=z$ 表示 d 的真实程度,$(0 \leqslant z \leqslant 1)$。

定义 1 将大数据的规模庞大、时效性强、类型多样化、准确性弱等特征分别通过 $|D|$、时间间隔函数 $s:D\rightarrow R\times R$、类型函数 $type:D\rightarrow T$、真实度函数 $v:D\rightarrow[0,1]$ 等加以体现。

2.2 大数据的分类

根据呈现特征的情况可把数据库向大数据的演变过程描述如图 1 所示。

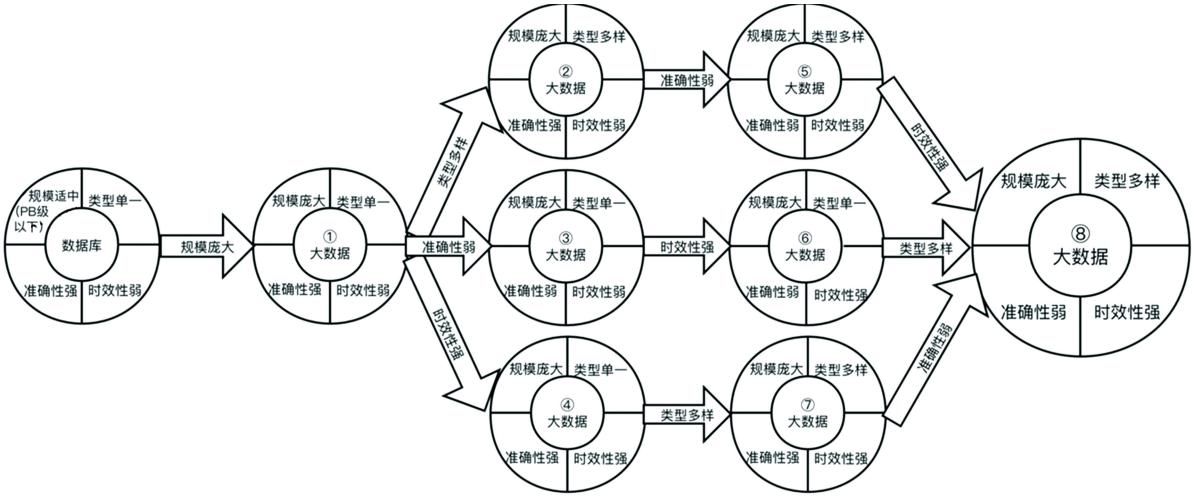

图 1 数据库向大数据的演变过程

Fig. 1 The evolution of the transformation from database to big data

图 1 中,数据库规模增长到 PB 级别后,演变为具有规模庞大、类型单一、准确性强、时效性弱的第一类大数据。

对于第一类大数据,保持规模庞大、准确性强、时效性弱不变的情况下,经过类型多样化后演变为第二类大数据;在同样的情况下,经过降低准确性后,演变为第三类大数据;同理,经过增强时效性后演变为第四类大数据。对于第二类大数据,保持规模庞大、类型多样性、时效性弱不变,经过降低准确性后演变为第五类大数据。对于第三类大数据,保持规模庞大、类型单一、准确性弱不变,经过增强时效性后演变为第六类大数据。对于第四类大数据,保持规模庞大、时效性强、准确性强不变,经过类型多样化后,演变为第七类大数据。第五类大数据经过增强准确性后,演变为第八类大数据;第六类大数据经过类型多样化后,演变为第八类大数据;第七类大数据经过减弱时效性强后,演变为第八类大数据。例如:在智能防疫系统中的推荐算法,基于用户定位、语音、检测时间等,对用户实时推送当地疫情防控政策,此类大数据具备规模庞大、类型多样、时效性强、准确性弱的特征,属于第八类大数据。

2.3 关系型大数据

第一类大数据中数据类型为关系型的大数据,称为关系型大数据,关系型大数据的形式化定义如下:

定义 2 关系型大数据 R_B 是一个形如$(D_R, T, type, s, v)$ 的 5 元组,其中:

(1) $D_R=\{r_1,\cdots,r_n\}$ 是一个元组集合(关系型大数据中数据对象为元组),$|D_R|$ 至少是 PB 级别;

(2) $T=\{\bar{a}:\bar{a}=(a_1,\cdots,a_k)\}$ 是一个数据库类型集合,$\bar{a}$ 描述了数据类型单一这一特征;

(3) $type:D_R\rightarrow T$ 是一个类型函数,对于元组 $r_i \in D_R(1 \leqslant i \leqslant n)$, $type(r_i)=\bar{a}$;

(4) $s:D_R\rightarrow\{(\infty,\infty)\}$ 是一个时间间隔函数,表示 r_i 的有效时间,表示时效性弱;

(5) $v:D_R\rightarrow\{1\}$ 是一个真实度函数,对于一个元组 $r_i \in D_R(1 \leqslant i \leqslant n)$, $v(r_i)=1$ 表示 r_i 的真实程度为 100%。

定义 2 中的数据类型仅为关系型,关系型大数据的规模庞大、时效性弱、准确性强等特征分别用 $|D_R|$、时间间隔函数 $s:D_R\rightarrow\{(\infty,\infty)\}$、真实度函数 $v:D_R\rightarrow\{1\}$ 等描述。

从上述定义可以看出,关系型数据是大数据的一种特例,因此数据库领域中的结论和研究方法可以拓展到大数据研究中。

3　关系型大数据查询的层次结构

在关系数据库中，查询是一个定义域为数据库，值域为关系的函数[3]；查询语言（query language）是描述查询的工具。为了满足应用需求，人们提出了多种具有特色的查询语言。如：Codd[15]的一阶关系演算和关系代数，Aho 等[16]的带有最小不动点算子的关系代数等。但对于查询问题的研究，需要统一的描述工具和框架。AK Chandra 等[3]给出了关系数据库查询的形式化定义，并以一阶语言（first-order language）为描述工具，提出了查询的层次结构，该层次结构涵盖了 Codd、Aho 和 Ullman 等查询语言所描述的查询，成为查询问题研究中查询分类的重要基础。

3.1　关系型大数据查询分类

AK Chandra 和 David Harel 用一阶语言为工具对查询进行描述，并利用¬（negation）、∘（composition）及 fixpoint 等运算，构成了查询的不动点查询层次（fixpoint query hierarchy）。不动点查询层次是关系数据查询的经典分类架构，关系型大数据查询与经典关系数据查询的区别在于数据规模，而与数据规模对查询描述语言和查询结构无关。因此，不动点查询层次也适用关系型大数据查询。不动点查询层次包含一阶查询集合 F、存在性查询集合 E，以及不动点查询集合 FP。FP 是一阶查询集合 F 和存在性查询的集合 E 在∘和 Y 运算下的闭包[3]。

令 L 为无函数且仅以 =（equality）、R_1、R_2… 为谓词符号的一阶语言；*First* 为由形如 $\bar{\boldsymbol{x}}$、$\bar{\boldsymbol{R}}$、$\boldsymbol{\Phi}$ 表达式构成的语言，其中 $\boldsymbol{\Phi}$ 是一阶语言 L 的公式，$\bar{\boldsymbol{x}}$ 是互不相同的变量组成的向量（仅包含在 $\boldsymbol{\Phi}$ 中出现的所有自由变量），$\bar{\boldsymbol{R}}$ 是一个互不相同的谓词符号组成的向量（包含所有出现在 $\boldsymbol{\Phi}$ 中的谓词符号）。

用 *First* 中的表达式可以描述一个查询。下列表达式描述的查询含义是选择 R_1 和 R_2 中具有相同 z 值的元组，其 x 和 y 值组合的二元组作为查询结果。

$$(x,y),(R_1,R_2),(\exists z)(R_1(x,z)\wedge R_2(z,y))$$

在实际应用中，R_1 和 R_2 可拓展为更多的属性。对于 $A\in First$，称由 A 表示的查询为一阶查询（记为 Q_A），称集合 $Q_{First}=\{Q_A\mid A\in First\}$ 为一阶查询集合（记为 F）。当数据规模达到 PB 级别以上，查询对象为关系型大数据时，具有上述结构的关系型大数据查询是一阶查询。

令 *Exist* 表示具有如 $\bar{\boldsymbol{x}}$，$\bar{\boldsymbol{R}}$，$(\exists\bar{y})\boldsymbol{\Phi}$ 形式的表达式集合，其中 $\boldsymbol{\Phi}$ 不含有量词。对于 $B\in Exist$，称由 B 表示的查询为存在性查询（记为 Q_B），称集合 $Q_{Exist}=\{Q_B\mid B\in Exist\}$ 为存在性查询的集合（记为 E）。当数据规模达到 PB 级别以上，查询对象为关系型大数据时，具有上述结构的关系型大数据查询是存在性查询。

First 和 *Exist* 是描述查询的基础语言，在其上应用¬（negation）、∘（composition）及 fixpoint 等运算，可获得更复杂的语言，相对应可以描述更复杂的查询。

其中，¬（negation）操作是对于一阶查询表达式 *First* 中描述的包含 $\bar{\boldsymbol{x}}$，$\bar{\boldsymbol{R}}$，$\boldsymbol{\Phi}$ 形式的查询表达 A，其否定形式 $\neg A$ 是形如 $\bar{\boldsymbol{x}}$，$\bar{\boldsymbol{R}}\neg\boldsymbol{\Phi}$ 的查询表达式。令 $A=\bar{\boldsymbol{x}}$，$(T_1,\cdots,T_n)\boldsymbol{\Phi}$，其中 T_i 的秩是 a_i，令 $\bar{C}=(C_1,\cdots,C_n)$，$C_i=\bar{y}_i\cdot(R_1,\cdots,R_k)\cdot\boldsymbol{\Psi}_i$，$|\bar{y}_i|=a_i$，用 $A\circ\bar{C}$ 来表示 $\bar{\boldsymbol{x}}\cdot(R_1,\cdots,R_n)\cdot\boldsymbol{\Phi}'$，其中，$\boldsymbol{\Phi}'$ 是用 $\boldsymbol{\Psi}_i$ 同时置换 $\boldsymbol{\Phi}$ 中的 T_i，按照出现顺序正确地重命名被约束或者相等的变量，使得 $\bar{y}_i$ 与 T_i 匹配。

令 LY 为按如下规则生成的公式集合：

（1）一阶语言 L 的公式属于 LY；

（2）令 $\boldsymbol{\Phi}\in LY$，秩为 a 的谓词 R 在 $\boldsymbol{\Phi}$ 中正出现（即 R 在偶数次¬运算的前提下自由出现），$(\bar{z},YR(\bar{\boldsymbol{x}}))\boldsymbol{\Phi}\in LY$，其中 Y 为最小不动点运算符，$\bar{z}$、$\bar{\boldsymbol{x}}$ 为不同变量构成的 $\bar{a}$ 元组；

（3）LY 在运算符∧、∨、¬及量词∀和∃下是封闭的。

基于 LY 中的公式构造出公式集合 *Fixpoint*，对于 *Fixpoint* 的公式 $\bar{\boldsymbol{x}}$，$\bar{\boldsymbol{R}}$，$\boldsymbol{\Phi}$，$\boldsymbol{\Phi}$ 是 LY 中的公式，$\bar{\boldsymbol{x}}$ 是一个由不同变量组成的向量，其中至少有一个 $\boldsymbol{\Phi}$ 中的自由变量，$\bar{\boldsymbol{R}}$ 是一个由不同谓词符号组成的向量，其中至少有一个 $\boldsymbol{\Phi}$ 中的自由谓词符号。

对于 $C\in Fixpoint$，称由 C 表示的查询为不动点查询（记为 Q_C），称集合 $Q_{Fixpoint}=\{Q_C\mid C\in Fixpoint\}$ 为不动点查询的集合（记为 FP）。当数据规模达到 PB 级别以上，查询对象为关系型大数据时，具有上述结构的关系型大数据查询是不动点查询。用 FP 中的表达式可以描述一个查询：

令 $\boldsymbol{\Phi}$ 为 $x=y\vee(\exists z)(R(x,z)\vee R'(y,z))$、$A=(x,y)\cdot(R,R')$，$\boldsymbol{\Phi}$，$K=(x,y)\cdot R((x,y)\cdot YR'(x,y))\boldsymbol{\Phi}$，$A$ 和 K 是 FP 中的表达式。K 是 A 的一个不动点，并且集合 Q_{YA} 表示为 TC 包含了自反传递闭包

Q_K。在关系型大数据查询中,不动点查询也可以用来表示嵌套和递归关系。

3.2 不同类别关系型大数据查询的预处理

本文将以不动点查询层次为基础,对于关系型大数据查询中的一阶查询、存在性查询、不动点查询研究其预处理问题并进行实验。

在关系型大数据查询中的一阶查询,经常存在如 $(x,y)\cdot(R_1,R_2)\cdot(\exists z)(R_1(x,z)\wedge R_2(z,y))$ 形式的查询,选择 R_1 和 R_2 中具有相同 z 值的元组这一操作在 SQL 中转化为连接操作。假如:R_1 与 R_2 的元组数均为 1 000 万条,未经任何预处理对关系型大数据进行一阶查询时,计算机会对两个表中的数据进行笛卡尔积,并从中间表中找出符合条件的结果输出,中间表扫描的次数会达到 10^{14} 次,即此关系型大数据的 $|D_R|$ 会随着数据量的增加迅速变大,使得查询求解变得困难。此时,若对关系型大数据的查询对象进行预处理,找到其核心,使大数据变小。对于 SQL 语言来说,可以将 WHERE 条件用 ON 进行描述,或者将查询拆解,把过滤后的数据通过简单一阶查询用视图存储在一张临时表中,再将原查询表与此临时表进行关联查询求出最后的结果。假如过滤出符合要求的元组数量有 10 万,那么预处理过后的查询,操作的关系型大数据其数据对象 $|D_R'|$ 至多为 10^{12},即新的查询是在其查询对象上关于 10^{12} 规模独立的[10]。

本文在内存 16 GB、固态硬盘 1 TB、MySQL 5.8 的计算环境下,并在加入索引后对一阶查询进行实验。实验对比了原查询与经过预处理后,在不同数量级下的查询解答耗时(查询解答时间的单位为 s),其结果如图 2 所示。

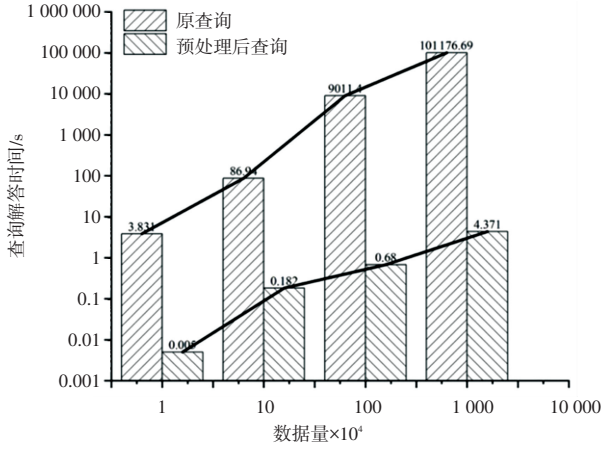

图 2 一阶查询解答时间变化趋势

Fig. 2 Changing trend of query answering time for first order query

由图 2 可知,在对原查询进行预处理后,查询解答时间明显缩短,并且当数据规模量级变大时,经过预处理后的查询其耗时增长平稳,均在多项式时间范围内。

关系型大数据查询中的存在性查询,经常存在关键字匹配。例如:使用 like 关键字检索,但其查询解答的性能较差,因为在使用 like 关键字检索时,会进行全表扫描来匹配字符串,对于关系型大数据来说,时间代价很大。在此场景下,可以引入倒排索引技术来进行预处理。此时,除了需要查询的表外,还需添加一张关键字表。关键字表包含字段关键字 id、关键字,并且对关键字这一列添加索引,再添加一张关联表把关键字表和需要查询的表关联,关键字 id 和查询表的主键作为关联表的联合主键。使用相同的实验方法,对此查询进行预处理前后的查询时间对比,其结果如图 3 所示。

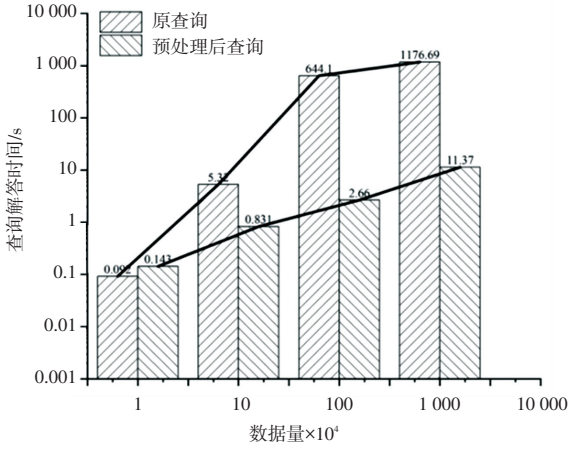

图 3 存在性查询解答时间变化趋势图

Fig. 3 Changing trend of query answering time for existence query

由图 3 可知,此预处理方法在数据量不大时效果并不明显,当数据量增大到 10 万甚至更大量级时,预处理对查询解答时间的优化非常明显。在此预处理方法中,使用倒排索引这一预处理优化技术,将关系型大数据查询中的数据部分新增了中间数据部分,虽然增加了存储空间与查询次数,但每一步操作都会命中索引,免去了全表扫描,是多项式时间内的,使关系型大数据查询得到了优化。一些大数据查询系统中也用到了这一关键技术,如 Elasticsearch 这一分布式、RESTful 风格的搜索和数据分析引擎等。

对于关系型大数据查询中的不动点查询,会频繁使用 union all 进行组合与嵌套,此时可以利用视图和预编译技术来进行预处理,减少查询解答时间。由于不动点查询是在一阶查询与存在性查询的基础

上，经过多种算子操作生成，所以可以充分利用一阶查询与存在性查询上的预处理方法，使得在关系型大数据上的查询解答时间缩短。

4　结束语

本文从大数据特征这一角度出发，对关系型大数据查询解答问题进行研究。给出了大数据和关系型大数据的形式化定义，描述了数据库向大数据的演变过程，归纳了 8 类大数据。对于不动点查询层次结构中的每一类查询进行论证，并举例讨论了其预处理方法。实验展示了预处理前后不同类型查询在大数据环境下，查询解答时间随数据量变化的趋势。

本文的研究工作为大数据应用提供了理论上的支持，后续工作将深入探讨关系型大数据的其他特征对查询解答的影响。

参考文献

[1] FAN W，GEERTS F，NEVEN F. Making queries tractable on big data with preprocessing：through the eyes of complexity theory [J]. Proceedings of the VLDB Endowment，2013，6(9)：685-696.

[2] 李建中，李英姝.大数据计算的复杂性理论与算法研究进展[J]. 中国科学：信息科学，2016，46(9)：1255-1275.

[3] CHANDRA A，HAREL D. Structure and complexity of relational queries[J]. Journal of Computer and System Sciences，1982，25(1)：99-128.

[4] SARMA A D，AFRATI F N，SALIHOGLU S，et al. Upper and lower bounds on the cost of a map-reduce computation [J]. Proceedings of the VLDB Endowment，2013，6(4)：277-288.

[5] AFRATI F N，ULLMAN J D. Optimizing joins in a map-reduce environment[C]//Proceedings of the 13th International Conference on Extending Database Technology. 2010：99-110.

[6] AFRATI F，DOLEV S，KORACH E，et al. Assignment problems of different-sized inputs in MapReduce[J]. ACM Transactions on Knowledge Discovery from Data (TKDD)，2016，11(2)：1-35.

[7] GAO Y，ZHANG Y，WANG H，et al. A distributed load balance algorithm of MapReduce for data quality detection[C]//Database Systems for Advanced Applications：DASFAA 2016 International Workshops：BDMS，BDQM，MoI，and SeCoP. Dallas，TX，USA：Springer International Publishing，2016：294-306.

[8] SARTHI P，RAJAN K，LAL A，et al. Generalized {Sub-Query} Fusion for Eliminating Redundant {I/O} from {Big-Data} Queries[C]//Proceedings of the 14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 20). 2020：209-224.

[9] ANUJA S，MALATHY C. Big data query optimization-literature survey[J]. 2021.

[10] FAN W，HUAI J P. Querying big data：bridging theory and practice[J]. Journal of Computer Science and Technology，2014，29(5)：849-869.

[11] LI J. Complexity，algorithms and quality of big data intensive computing [C]//Proceedings of the Database Systems for Advanced Applications-19th International Conference，DASFAA. 2014.

[12] GAO X，LI J，MIAO D，et al. Recognizing the tractability in big data computing[J]. Theoretical Computer Science，2020，838：195-207.

[13] FAN W，GEERTS F，CAO Y，et al. Querying big data by accessing small data [C]//Proceedings of the 34th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems. 2015：173-184.

[14] ZHANG A Z，LI J Z，GAO H. Interval estimation for aggregate queries on incomplete data[J]. Journal of Computer Science and Technology，2019，34(6)：1203-1216.

[15] BHOSALE S T，PATIL T，PATIL P. Sqlite：Light database system[J]. Int. J. Comput. Sci. Mob. Comput.，2015，44(4)：882-885.

[16] HAREL D. Computable queries for relational data bases[J]. Journal of Computer and System Sciences，1980，21(2)：156-178.