基于MapReduce框架的海量数据相似性连接研究进展

庞 俊[1] 于 戈[1] 许 嘉[2] 谷 峪[1]
(东北大学信息科学与工程学院 沈阳 110819)[1] (国防科学技术大学信息系统与管理学院 长沙 410073)[2]

摘 要 海量数据相似性连接作为海量数据处理的基本操作,在文本聚类、剽窃检测、实体解析等研究领域具有重要作用。另一方面,MapReduce编程模型因为具有良好的可扩放性、容错性和易用性,被广泛地应用于海量数据处理。因此,基于MapReduce框架的海量数据相似性连接查询技术成为海量数据处理领域的热点问题之一。首先,概括了海量数据固有特点和MapReduce编程框架的缺陷给现有相似性连接查询技术带来的巨大挑战;其次,提出了海量数据相似性连接的定义,按3种不同的分类标准对其进行了分类;接着,重点分析了集合、字符串和向量数据类型的海量相似性连接查询最新技术,并从效率和适用范围等方面分别对这些技术进行了比较;最后,讨论了海量数据相似性连接查询技术亟待解决的关键问题,并提出了一些有前景的解决方案。

关键词 海量数据,相似性连接,MapReduce,Top-k

中图法分类号 TP391 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.1.001

Similarity Joins on Massive Data Based on MapReduce Framework

PANG Jun[1] YU Ge[1] XU Jia[2] GU Yu[1]
(School of Information Science and Engineering, Northeastern University, Shenyang 110819, China)[1]
(School of Information System and Management, National University of Defense Technology, Changsha 410073, China)[2]

Abstract As a basic operation of large-scale data processing, similarity joins on large-scale data play an important role in document clustering, plagiarism detection, entity resolution and many other fields. On the other hand, the MapReduce programming model is widely applied to massive data processing because of its good scalability, fault tolerance and usability. Therefore, similarity joins on massive data based on MapReduce become one of the hot topics in the field of massive data processing. Firstly, big challenges of similarity join query introduced by rapid growth of data volume were generalized. Then, the definition of similarity joins on massive data was presented and similarity joins on massive data were classified according to three different standards. In addition, the current status of set, string and vector similarity joins on massive data were emphatically analyzed and compared from the aspects of efficiency, applicability and so on. Finally, the research focus and trend of this area were investigated and the promising solutions were suggested.

Keywords Massive data, Similarity join, MapReduce, Top-k

1 引言

海量数据一般指TB级及其以上的数据。海量数据不但普遍存在,而且快速增长。在学术界和工业界可以产生超大规模的数据。例如:2000年斯隆数字巡天项目启动时,新墨西哥周的天文望远镜在短短几周内收集到的数据,已超过天文学历史收集数据的总和[1]。脸谱网注册用户超过10亿,每月上传的照片超过10亿张,每天生成300TB以上的日志数据[2]。据互联网数据中心(Internet data center, IDC)预计,全球数据总量将在2020年达到35ZB,是2010年1.2ZB的29倍[3]。

海量数据管理和应用可以为企业带来巨额的经济利益。自2005年以来,IBM投资160亿美元完成了30次与海量数据有关的收购,促使其业绩稳定高速增长。2012年,IBM股价突破200美元大关,3年之内翻了3番[2]。eBay通过挖掘海量数据来优化广告投放,从而使产品广告费自2007年以来降低了99%,顶级卖家占总销售额的百分比上升至32%[2]。

海量数据相似性连接查询,即从海量数据源中检索出所有的相似对象对,作为海量数据处理的一项基本操作,受到广泛的关注[4-8]。在文档聚类中,使用相似性连接技术,可以自动通知网页的URL已发生改变并且找到新的URL,还有助于广泛地更新分布式信息[4];在剽窃检测中,使用相似性连接技术,可以检测来源相同的文档,还可以识别剽窃物、修订版本和文档的不同版本[5]。

到稿日期:2014-01-10 返修日期:2014-05-05 本文受国家重点基础研究发展计划(973)项目(2012CB316201),国家自然科学基金(61272179,61173028),教育部博士点基金(20120042110028),教育部-英特尔信息技术专项科研基金(MOE-INTEL-2012-06)资助。

庞 俊(1983-),男,博士生,CCF学生会员,主要研究方向为云计算和相似性连接,E-mail:pangjun@research.neu.edu.cn;于 戈(1962-),男,博士,教授,博士生导师,主要研究方向为数据流、数据挖掘、分布式数据库,E-mail:yuge@mail.neu.edu.cn(通信作者);许 嘉(1984-),女,博士,主要研究方向为RFID数据管理和相似性连接;谷 峪(1981-),男,博士,副教授,主要研究方向为空间数据管理和图数据管理。

MapReduce 编程框架[9]已被国内外很多知名企业用来处理海量数据，比如：谷歌、脸谱网、亚马逊和阿里巴巴。基于MapReduce模型的海量数据相似性技术成为海量数据处理领域的热点问题之一[14-24]。但是海量数据固有的特点以及MapReduce编程框架的缺陷给现有的相似性连接技术带来了巨大的挑战：

(1)大规模性和快速增长性。海量数据不但规模庞大，而且增长快速。首先，海量数据的存储是一个巨大的挑战，因为现实生活中产生的海量数据的规模已达到了PB级[1]，远远超过了关系数据库的最大存储量(以TB计量)。其次，现有的相似性连接查询技术不适用于海量数据相似性连接查询，因为这些技术绝大部分是集中式内存算法或者外存算法，处理效率不理想。最后，并行数据库技术也不适合用来处理海量数据[10]，因为其可扩放性(Scalability)和非事务级容错能力不佳。

(2)多模态和复杂性。相似性连接查询需要使用多种相似性度量函数[11]来处理多种数据类型，比如：集合、字符串和向量等。因此，要求相似性连接查询算法具有处理多模态数据的能力。海量数据包含结构化数据、半结构化和非结构化数据[12]。与结构化数据相比，半结构化和非结构化数据的相似性连接查询技术不够成熟，面临严峻的挑战。

(3)实时性和自适应性。现实生活中很多数据具有时效性，需要快速处理。例如：电子商务数据需要实时分析，以便及时做出补货等决策。海量数据时代，企业业务需求的快速更新，要求相关海量数据的分析和处理模型具有快速适应新业务需求的能力[13]。

而另一方面，流行的MapReduce编程框架在处理复杂海量数据时也存在一些不足：a)标准MapReduce算法的效率易受输入数据分布影响(不均匀的数据分布，引起MapReduce节点的负载不均衡，从而降低算法的执行效率)；b)标准的MapReduce框架进行单数据源(单数据集或表)操作的效率比较高，进行多数据源(多个数据集或表)操作的效率比较低；c)MapReduce框架不适合进行海量数据的实时处理，而常被用来进行离线处理。

本文第2节对相似性连接查询进行定义和分类；第3节—第5节分别综述了基于MapReduce的集合、字符串和向量的相似性连接查询的研究进展；第6节探讨了海量数据相似性连接查询技术未来的发展趋势；最后总结全文。

2 定义和分类

2.1 定义

相似性连接查询技术已取得了很多创新性成果[14-40]。关于相似性连接查询，国内外文献尚无统一定义。例如文献[26]定义了空间相似性连接查询：(1)自连接：已知一个由N个高维点组成的集合和一个距离度量，从该集合中找出所有距离不超过阈值ε的点对；(2)非自连接：已知由高维点组成的两个集合和一个距离度量，找出所有满足以下条件的点对：两个点分别来自这两个集合，且距离不超过阈值ε。文献[32]定义了Top-k相似性连接查询：已知两个由记录组成的集合和相似性函数，找出分别来自这两个集合的最相似的k个记录对。

海量数据相似性连接是指：从一个或两个以上(含两个)规模巨大(超过TB)的数据源中找出所有相似二(或多)元组。相似二(或多)元组的对象来自同一个数据源或者分别来自不同的数据源。具体地，基于阈值的“相似”是指：通过相似性函数(距离函数)计算出的相似度不小于相似性阈值(距离阈值)；Top-k“相似”是指根据相似性函数(或距离函数)计算出的相似度(或距离)按降序(或升序)排序排在前k位。

2.2 分类

海量数据相似性连接查询按照不同的分类标准，分成以下3种主要类别：(1)按照“相似”定义的不同，可分为阈值连接查询和Top-k连接查询；(2)按照数据源数目的不同，分为单源(自连接)、双源和多源连接查询，单源连接查询是指找出的所有相似二元组的对象来自同一个数据源；(3)按照数据类型的不同，可分为集合、字符串、向量和图等连接查询。集合连接查询处理对象的数据类型是集合。不同类别的相似性连接查询可以相互交叉，比如：阈值连接查询，既可以是单源连接查询，也可以是双源连接查询。下面主要分析和比较集合、字符串和向量这3种常见数据类型的海量数据相似性连接最新技术。

3 海量集合相似性连接查询

海量集合相似性连接方法一般利用倒排索引或者副本对原数据集进行分组，然后在各组内部分别进行连接操作并聚集，从而得到最终结果。为了提高算法的效率，对算法进行了优化，比如，使用前缀过滤技术。

文献[15]提出了基于MapReduce和过滤技术的方法，即使用集合杰卡德相似性函数完成单源或双源的基于集合相似性的记录连接查询。其基本思想是：利用相似的两个集合至少有一个公共元素这个特征，建立倒排索引，对原数据集进行分组，保证组内的集合可能相似，组间的集合必不相似。然后，并行处理这些分组，计算和整合部分结果，得到最终结果。该文首次利用MapReduce框架来解决海量集合数据相似性连接查询问题，并且做了以下两点工作来提高算法效率：(1)在Map阶段使用前缀过滤技术，在Reduce阶段使用长度、位置和后缀过滤技术，减少了副本数和计算次数；(2)按词频对所有集合元素进行升序排序，这不但可以增强前缀过滤的效果，而且可以在一定程度上保证Reducer的负载均衡。该文提出的方法适用于所有常用集合相似性度量函数，也适合处理基于字符串相似性的记录连接问题。

整个方法分3步：1)统计所有集合元素的频率，并按升序排序；2)过滤得到候选集合，并验证获得相似集合ID对；3)去掉重复结果并连接相似集合对。每步分别提出两种方法，因此可得到多个集合相似性连接算法。其中，BTO-PK-OPRJ方法最好，但是只适用于处理较小规模的单、双源数据集。BTO-PK-BRJ方法仅次于BTO-PK-OPRJ方法，而且不局限于较小规模数据集。但是，BTO方法扩展性不好，因为需要对所有Token进行集中排序和分组，所以不适用比较大的Token字母表。另外，PK方法在Map阶段产生了原始集合数据的多个副本，造成较大的磁盘I/O代价和通信代价。并且，PK方法还存在多次重复计算问题和负载不均衡问题。

文献[16]提出了几种基于MapReduce的通用算法，即使用集合杰卡德相似性函数、汉明距离函数或者字符串编辑距离函数完成单源集合、位字符串或字符串的相似性连接查询。

其基本思想是：通过副本把原始数据分成若干个组；然后并行处理各组获得的部分结果，聚集所有部分结果，得到最终结果；同时要求副本数尽可能少，并且无重复计算和不输出重复结果。

该文提出了基本(Naive)、球哈希 I (Ball-hashing-1)、汉明编码(Hamming code)等 6 个算法，其适用于多种数据类型(集合、位字符串和字符串)数据的连接查询操作。所有算法都只需要一个 MapReduce 作业。该文还提出了一个理论代价模型，首次从理论上对这些 MapReduce 通用连接算法进行了比较；使用了以下衡量标准：Map 阶段所有 Mapper 的计算总代价、Shuffle 阶段的总通信代价、Reduce 阶段所有 Reducer 的计算总代价和 Reducer 数目。

此外，该文提出的理论模型和进行的理论分析都建立在一些理论假设之上。比如：数据均匀分布。而现实情况不一定如此。为了方便比较，该文提出的方法几乎没有结合过滤技术。其中，基本方法产生 Reducer 的数量很大，可能引起巨大的磁盘 I/O 代价和通信代价。球哈希 I 方法不但产生很多副本，而且产生多个不需要进行任何处理的 Reducer。

基于 MapReduce 的海量集合、字符串和向量相似性连接查询算法在过滤技术、可适用数据类型等方面的比较结果如表 1 所列。表 1 的第三列表示算法是否精确，第四列的“集、字、位字和向”分别表示“集合”、“字符串”、“位字符串”(比特字符串)和“向量”；第五列表示属于阈值连接或者 Top-k 连接；倒数第一列“可扩放性”指平均延迟可扩放性；倒数第一列和第二列的“未比较”表示原文中未进行相关比较，因此没有结论。

表 1　基于 MapReduce 的海量数据相似性连接查询算法

文献	算法	是否精确	数据类型	连接类型	数据源数	过滤技术	相似性度量函数	运行时间	可扩放性
[15]	BTO-PK-OPRJ	是	集或字	阈值	1 个或 2 个	前缀、长度、位置和后缀过滤	集合杰卡德相似性	小规模数据集，比 BTO-PK-BRJ 快	比 BTO-PK-BRJ 差
[15]	BTO-PK-BRJ	是	集或字	阈值	1 个或 2 个	前缀、长度、位置和后缀过滤	集合杰卡德相似性	较大规模数据集，比 BTO-PK-OPRJ 快	比 BTO-PK-OPRJ 好
[16]	Naive	是	集、位字或字	阈值	1 个	无	汉明距离、编辑距离或集合杰卡德相似性	比 BTO-PK-BRJ 快	未比较
[16]	Ball-hashing-1	是	集、位字或字	阈值	1 个	无	同 Naive 算法	未比较	未比较
[16]	Ball-hashing-2	是	集、位字或字	阈值	1 个	无	同 Naive 算法	未比较	未比较
[16]	Splitting	是	集、位字或字	阈值	1 个	分离重叠过滤	同 Naive 算法	比 Naive 快	未比较
[16]	Anchor Points	是	集、位字或字	阈值	1 个	无	同 Naive 算法	比 Naive 好	未比较
[18]	Online-aggregation	是或否	集、字或向	阈值	1 个	重叠过滤	与对象数据类型相对应	比 Lookup 和 Sharding 快	未比较
[18]	Lookup	是或否	集、字或向	阈值	1 个	重叠过滤	与对象数据类型相对应	小规模数据集，比 Sharding 快	未比较
[18]	Sharding	是或否	集、字或向	阈值	1 个	重叠过滤	与对象数据类型相对应	比 BTO-PK-BRJ 快	未比较
[17]	MRSimJoin	是	度量空间(向、字)	阈值	1 个或 2 个	阈值划分过滤	与对象数据类型相对应	比 1-bucket-theta 快	接近线性
[14]	SSJ-2	是	向	阈值	1 个	前缀过滤	规范化的向量余弦相似性	比 BTO-PK-BRJ 方法快	未比较
[14]	SSJ-2R	是	向	阈值	1 个	前缀过滤	规范化的向量余弦相似性	比 SSJ-2 快	未比较
[20]	TopK-FB-MR	是	向	Top-k	1 个	阈值过滤	欧几里德距离或闵可夫斯基距离	未比较	接近线性
[20]	TopK-FT-MR	是	向	Top-k	1 个	阈值过滤	欧几里德距离或闵可夫斯基距离	比 TopK-FB-MR 快	接近线性

4　海量字符串相似性连接查询

海量字符串相似性连接查询使用 k 中心点和倒排索引对原数据集进行分组，然后求解问题；并利用迭代划分的思想、全过滤技术以及相似度计算公式的特点，减少计算代价和通信代价，改善算法的性能。

文献[17]提出基于 MapReduce 的迭代划分方法(MR-SimJoin)，完成了单源或双源度量空间(Metric space)的相似性连接查询。其基本思想是：对原始数据集进行迭代划分，使划分获得的所有数据集足够小；然后使用高效的内存相似性连接算法处理各划分，获得部分结果；最后汇总部分结果获得最终结果。

该文首次结合 MapReduce 框架和迭代划分思想，使用多个作业来完成相似性连接查询。MRSimJoin 方法简单易懂，适用于度量空间的所有数据类型(比如：向量和字符串)的相似性连接查询。计算每个小划分得到的部分结果中无重复相似对象对。同时，MRSimJoin 方法在划分时采用划分阈值过滤技术，减少了划分规模，提高了算法效率。该方法比文献[21]的 1-bucket-theta 方法好，表现在：运行速度更快(见表 1)，处理高维向量时更稳定(随着维度的增加，执行时间先减少，然后趋于不变)。MRSimJoin 算法存在一些不足：每次划分所取 k 值都相同。实际上，取不同的 k 值可能获得更好的效率[41]。该文没有将 MRSimJoin 算法与文献[15]提出的 BTO-PK-BRJ 算法进行比较。

文献[18]提出基于 MapReduce 和过滤技术的两步方法，来完成单源多重集合、集合、字符串或者向量的相似性连接查询。其基本思想是：多重集合对的相似性计算可以通过各多重集合的单方面函数(Unilateral function)和它们的连接函数(Conjunctive function)的计算获得。单方面函数的计算只需要扫描多重集合。连接函数的计算只需要扫描多重集合的交集。因此，先计算多重集合的单方面函数值和它们的连接函数值，然后计算相似度，获得最终结果。

该文提出的通用算法适用于多种数据类型的数据的相似性连接查询和多种相似性度量函数。这些算法使用了全过滤技术，提高了效率，处理多重集合或集合时执行速度比 BTO-PK-BRJ 方法快。

不失一般性，以多重集合的相似性连接查询为例介绍该文提出的算法。第一步，完成多重集合单方面函数的计算，并与多重集合各元素连接。这一步提出了 3 个算法：(1)Online-aggregation；(2) Lookup；(3) Sharding。算法(1)只支持 Google MapReduce。算法(2)第一个作业的结果要加载到内存中，因此它只适用于小规模数据集。处理小规模数据集时，执行效率降序排序如下：(1)＞(2)＞(3)＞BTO-PK-BRJ；处理大规模数据集时，执行效率降序排序如下：(1)＞(3)＞BTO-PK-BRJ。

第二步，建立倒排索引，计算候选结果的连接函数和相似度，获得最终结果。该步骤有 3 种方法。方法一的第一个作业利用上一步的结果，建立倒排索引，获得候选结果集，从而计算出所有可能相似的多重集合对。第二个作业验证第一个作业的结果。处理大规模数据集时，第一个作业获得的候选结果集可能很大，不能全部装载在内存中；方法二不计算大小超过指定参数的候选集(类似去掉“停用词”)；方法三把大小超过指定参数的候选集进行基本划分。方法一只适合处理小规模数据集。方法二和方法三分别是适用于大规模数据集的近似方法和精确方法。

5 海量向量相似性连接查询

海量向量相似性连接查询使用倒排索引对原单个数据集进行分组，利用前缀过滤、阈值过滤和相似性计算公式的特点提高算法的性能。

文献[14]提出 MapReduce 算法，其使用规范化的向量余弦相似性函数，完成单源文档的相似性连接查询。其基本思想是：用特征码(Signature)多重集合表示所有文档，统计所有特征码的全局频率，并按频率进行升序排序；然后建立前缀特征码倒排索引，过滤，从而获得候选集，嵌套循环处理各候选集(包括去重和验证)，获得部分结果，聚集所有部分结果，从而获得最终结果。

该文第一次结合 MapReduce 框架和过滤技术来完成海量文档的相似性连接查询，并且做了以下 4 点工作来提高算法效率：(1)使用了过滤效果优于全过滤技术[40]的前缀过滤技术；(2)使用文档 ID 表示文档，减少了通信代价；(3)有效利用前缀过滤剪掉的信息，减少了磁盘 I/O 代价；(4)使用分块(Blocking)技术，一定程度上保证了负载均衡。该文提出方法的运行速度不但比 BTO-PK-BRJ 方法快，而且比文献[40]提出的算法快。

整个方法分 3 个阶段：第一阶段，对原始文档集进行预处理，统计各特征码的频率并按升序排序；第二阶段，建立前缀特征码倒排索引，分割原始数据集；第三阶段，去掉重复的文档对，验证获得相似文档对。与文献[15]不同，该文先使用文档 ID 替代整个文档的内容，然后利用前缀过滤技术获得可能相似的文档 ID 对，接着访问分布式文件系统(Distributed file system，DFS)，获取各候选文档 ID 对对应的文档，验证出相似文档 ID 对。或者在获得候选文档 ID 对之后，通过前缀特征码传递过来的信息、前缀过滤技术剪掉的信息和原始数据集信息来验证出相似文档 ID 对。

这两种方法(SSJ-2 和 SSJ-2R)处理文档相似性连接的速度都比 BTO-PK-BRJ 方法快，并且 SSJ-2R 方法比 SSJ-2 方法更快。因为 SSJ-2 每计算一次文档对相似性就需要访问一次 DFS，产生非常大的磁盘访问代价。

SSJ-2R 方法存在一些不足：向量特征值采用 TF，而没有采用常用的 TF-IDF。该方法的局限性很强，很难扩展到非文档数据类型的相似性自连接查询。

文献[20]提出了 MapReduce 算法，即使用欧几里德距离函数完成单源向量 Top-k 相似性连接查询。其基本思想是：利用副本对原始数据集进行分组，分别求出每个组的 Top-k 相似向量对，对所有组的结果进行聚集和排序，获得原始数据集的 Top-k 相似向量对。

该文首次提出基于 MapReduce 和欧几里德距离的 Top-k 相似性连接算法，并使用阈值过滤技术来提高算法的效率：先采样，计算出 Top-k 相似性连接的阈值上限；然后利用该上限过滤掉相似度超过该上限的向量对，从而减少副本数和计算次数。这些算法不但适用于欧几里德距离，而且适用于闵可夫斯基距离。Top-k 连接不需要指定阈值，因此不用为了选择一个较好的阈值而进行多次试验。

此外，该文使用了一个 MapReduce 作业来完成 Top-k 相似性连接。Map 阶段完成数据的组合划分，Reduce 阶段完成各组合划分最相似的前 k 个结果的计算，主程序完成排序和获得最终结果。该文还提出了两种划分方法和 3 种内存 Top-k 相似性连接算法。这些算法又可组合成 6 种算法。其中，Top$K$-FT-MR 算法[20]执行速度最快，可扩放性(平均延迟)最好——接近线性。Top$K$-FB-MR 算法[20]稍逊。

Top$K$-FT-MR 算法使用了比较复杂的安全桶划分技术，存在重复计算问题。Top$K$-T 算法[20]也存在重复计算问题。用来保证给每个桶分配近似数量元素的合并直方图方法比较粗糙，取得的效果不理想，可能带来比较严重的负载不均衡问题。

6 亟待解决的关键问题

海量数据的相似性连接技术虽然已取得了一些创新性成果，但是尚有很多关键问题有待进一步研究：

(1)多源的海量数据相似性连接技术。据作者所知，多源海量数据相似性连接查询的研究尚未见报道。文献[14-20]研究了单源或双源集合、字符串和向量数据类型的相似性连接技术。多源的相似性连接虽然可以转化为多个双源相似性连接，但是直接进行多源连接的算法的效率可能更高[37]。因此，可以设计直接进行多源连接的算法，来缩短多源连接的执行时间。

(2)在线实时的海量相似性连接技术。目前，MapReduce 框架因为良好的可扩放性、容错性和易用性，被用来完成海量数据的相似性连接。不过，作为一种批处理模型，MapReduce 不适合用于实时处理。海量数据相似性连接的在线实时处理有待进一步研究。

(3)海量数据 Top-k 连接技术。海量数据 Top-k 连接查询不使用相似性阈值，因此适合在不容易指定相似性阈值的情形下使用。但是，据作者所知，这方面的研究很少(仅有文献[20]研究了海量向量的 Top-k 连接)。可以考虑结合 Map-

Reduce框架和小规模数据集 Top-k 连接技术，充分利用它们各自的优势，来完成海量数据的 Top-k 连接。

(4)海量 XML 文档等复杂数据结构的相似性连接技术。现有相关研究主要集中在集合、字符串和向量数据类型。但是，海量数据不局限于这 3 种数据类型，还有 XML 文档、数据流、不确定对象、图和直方图等。因为数据类型不同，所以现有集合、字符串和向量的相似性连接技术不一定适用于这些新型数据。对于这些新型数据的相似性连接技术的开发，可以通过以下两种方式实现：a)借鉴现有的集合、字符串和向量海量相似性连接技术；b)基于 MapReduce 框架，扩展高效的集中式算法。

(5)支持多种数据类型的海量数据相似性连接技术。现有的海量相似性连接技术大多适用于一种或两种数据类型，很少适用于 3 种或 3 种以上数据类型。文献[18]虽然研究了支持集合、字符串和向量数据类型的相似性连接技术，但是仅仅进行了初步尝试。现实生活中，存在多种数据类型。如果每种数据类型的连接处理都需要一种不同的算法，那么一个通用的系统实现的算法数量将与现实生活中存在的数据类型数量一样多。这将极大地增加系统开发的工作量。

(6)基于 MapReduce 的负载均衡的海量相似性连接技术。数据分布不均匀和非线性的 Reduce 端算法将引起和加深 Reduce 端的负载不均衡[42,43]。负载不均衡问题带来很多危害：增加了作业完成时间，不能最大化利用资源等。可是，现实生活中的数据往往分布不均匀。因此，负载不均衡问题的出现将不可避免。目前，解决该问题的技术比较简单，效果不理想。可以考虑借鉴并行、分布式数据库和其他 MapReduce 算法处理数据分布不均匀问题的方法，来解决相似性连接中遇到的负载不均衡问题。

结束语 本文在对相似性连接定义进行概括的基础上，提出了海量数据相似性连接的定义，归纳了海量数据相似性连接查询面临的主要挑战，仔细分析和比较了 3 种数据类型(集合、字符串和向量)的海量数据相似性连接查询的最新技术，最后提出了对海量数据相似性连接技术未来发展趋势的一些见解。

参 考 文 献

[1] Viktor M S, Kenneth C. Big data: A Revolution that will transform how we live, work and think[M]. 盛杨燕，周涛，译. 杭州：浙江人民出版社，2013：9-10

[2] 李国杰. 大数据研究的科学价值[J]. 中国计算机学会通讯，2012，9(8)：8-15

[3] Miller R. 'Digital universe' nears a zettabyte[EB/OL]. [2012-11-24]. http://www.datacenterknowledge.com/archives/2010/05/04/digital-universe-nears-a-zettabyte/

[4] Broder A Z, Glassman S C, Manasse M S, et al. Syntactic clustering of the web[J]. Computer Networks and ISCN Systems, 1997, 29(8-13): 1157-1166

[5] Hoad T C, Zobel J. Methods for identifying versioned and plagiarized documents[J]. Journal of the American Society for Information Science and Technology, 2003, 54(3): 203-215

[6] Kolb L, Thor A, Rahm E. Load balancing for MapReduce-based entity resolution [C]// Proc of ICDE. Washington: IEEE, 2012: 618-629

[7] Cho J, Shivakumar N, Garcia-Molina H. Finding replicated web collections [C]// Proc of SIGMOD. New York: ACM, 2000: 355-366

[8] 相似图片搜索引擎 TinEye [EB/OL]. [2012-11-24]. http://www.ipc.me/tineye.html

[9] Dean J, Ghemawat S. MapReduce: simplified data processing on large clusters[C]// Proc of OSDI. New York: ACM, 2004: 137-150

[10] Zhang X F, Chen L, Wang M. Efficient multi-way theta-join processing using MapReduce[J]. PVLDB, 2012, 5(11): 1184-1195

[11] 庞俊，谷峪，许嘉，等. 相似性连接查询技术研究进展[J]. 计算机科学与探索，2013，7(1)：1-13

[12] 智慧城市. "智慧"来自大数据[EB/OL]. [2012-11-24]. http://www.im2m.com.cn/107/10492963023.shtml

[13] CIO时代网. 浅析大数据的特点[EB/OL]. [2012-12-13]. http://www.ciotimes.com/baike/62989.html

[14] Baraglia R, Morales G D F, Lucchese C. Document similarity self-join with MapReduce [C]// Proc of ICDM. Washington: IEEE, 2010: 731-736

[15] Vernica R, Carey M J, Li C. Efficient parallel set-similarity joins using MapReduce [C]// Proc of SIGMOD. New York: ACM, 2010: 495-506

[16] Afrati F N, Sarma A D, Menestrina D, et al. Fuzzy joins using MapReduce[C]// Proc of ICDE. Washington: IEEE, 2012: 834-845

[17] Silva Y N, Reed J M. Exploiting MapReduce-based similarity joins [C]// Proceedings of SIGMOD. New York: ACM, 2012: 693-696

[18] Metwally A, Faloutsos C. V-SMART-Join: a scalable MapReduce framework for all-pair similarity joins of multisets and vectors[C]// Proc of VLDB. New York: ACM, 2012: 704-715

[19] Silva Y N, Reed J M, Tsosie L M. MapReduce-based similarity join for metric spaces[C]// Proc of Cloud-I. New York: ACM, 2012: 3

[20] Kim Y H, Shim K. Parallel top-k similarity join algorithms using MapReduce[C]// Proc of ICDE. Washington: IEEE, 2012: 510-521

[21] Okcan A, Riedewald M. Processing theta joins using MapReduce [C]// Proc of SIGMOD. New York: ACM, 2011: 949-960

[22] Zhang C, Li F F, Jestes J. Efficient parallel kNN joins for large data in MapReduce [C]// Proc of EDBT. New York: ACM, 2012: 38-49

[23] Lu W, Shen Y Y, Chen S, et al. Efficient Processing of k nearest neighbor joins using MapReduce [J]. PVLDB, 2012, 5(10): 1016-1027

[24] Yokoyama T, Ishikawa Y, Suzuk Y. Processing all k-nearest neighbor queries in hadoop[C]// Proc of WAIM. Berlin: Springer, 2012: 346-351

[25] Bentley J L, Shamos M I. Divide-and-conquer in multidimensional space[C]// Proc of STOC. New York: ACM, 1976: 220-230

[26] Shim K, Srikant R, Agrawal R. High-dimensional similarity joins [C]// Proc of ICDE. Washington: IEEE, 1997: 301-311

[27] Shafer J C, Agrawal R. Parallel algorithms for high-dimensional similarity joins for data mining applications [C]// Proc of VLDB. New York: ACM, 1997: 176-185

(下转第 27 页)

在仿真结束后，其判断依据为 $P15>P14$，若成立，则 P 为 true；否则 P 为 false。通过调整指标取值，更新模型元素和重复仿真运行，实现了综合合理性的评估。

表 3　设计指标与时效仿真结果对照表

指标名称	物理含义	组合 1	组合 2	组合 3
T_{comm}(s)	通信延时	0.2	0.1	0.1
T_{camer}(s)	照相工作周期	2	2	2
D_{camer}(s)	照相工作延时	0.6	0.4	0.4
$T_{control}$(s)	控制响应延时	0.2	0.1	0.1
s(m)	障碍物判断距离	15	15	10
l(m)	刹车制动距离	3	3	2
P	综合合理性	false	true	true

本文提出的基于扩展 DPN 语义的 CPS 混成行为时效建模与评估方法总体可概括为以下步骤：

(1)通过对信息感控过程进行抽象，依据其感知逻辑和时间特性建立扩展 DPN 模型；

(2)依据离散控制过程模型和连续物理规律方程，推导出关键控制量的最简状态方程，并据此建立其扩展 DPN 模型；

(3)通过分析 CPS 集成行为的物理控制效应目标，对各指标和参数合理性进行抽象和推导，确定合理性判断依据；在此基础上建立信息物理混成行为和相应时效合理性的集成模型；

(4)依据各指标与模型元素值的对应关系，调整指标参数和模型；通过重复仿真运行实现多指标综合合理性的评估。

在现有的带计算时间语义的 CPS 系统行为仿真技术中，PTIDEs 编程模型[6]为计算过程绑定计算时间，对计算行为的时序演化模拟提供了途径；基于 Modelica 及其嵌入式扩充库的信息物理集成计时仿真即为确定性耗时的计算过程与连续物理动态过程的集成仿真[7,8]；相比较而言，扩展 DPN 模型由于不具有灵活的赋值表达能力，不适于进一步生成可执行软硬件框架，因此只能用于性能评估；但基于扩展 DPN 模型的建模与仿真为各实时指标和感控参数的集成建模提供了更简洁且语义统一精确的数学模型，在多指标评估阶段避开了系统实体的程序设计，简化了评估过程。

结束语　本文阐述了扩展 DPN 模型的结构和执行语义，以某智能车自主行进与紧急避障过程为研究对象，建立了其关键信息过程、行进控制过程与路程积分过程的统一模型；通过分析和推导其系统控制物理目标，建立了该信息物理混合过程与物理时效的集成模型；通过调整指标参数及模型的仿真运行，实现了多指标的综合评估。该方法为信息物理混成行为集成时效合理性的评估及多指标的组合设计提供了一种简洁有效的途径。

参 考 文 献

[1] 何积丰. Cyber-physical systems[J]. 中国计算机学会通讯，2010,6(1):25-29

[2] Lee E A. CPS foundations[C]// Proc. of Design Automation Conference. 2010:737-742

[3] David R, Alla H. Discrete, continuous, and hybrid Petri nets [M]. Germany:Springer,2005

[4] Demongodin I, Koussoulas N T. Differential Petri Nets: Representing Continuous Systems in a Discrete-Event World[J]. IEEE Transactions on Automatic Control,1998,43(4):573-579

[5] Hybrid Petri Net ICSI Simulator[OL]. http://sourceforge.net/projects/hisim,2013

[6] Center for hybrid and embedded software systems. Timing-Centric Software [OL]. http://chess.eecs.berkeley.edu/ptides/,2010

[7] Henriksson D, Elmqvist H. Cyber-physical systems modeling and simulation with modelica[C]// Proceedings 8th Modelica Conference. 2011:502-509

[8] Malmheden M, Elmqvist H, et al. ModeGraph-A Modelica Library for Embedded Control Based on Mode-Automata[C]// Proceeding of 6th International Modelica Conference. 2008:255-267

(上接第 5 页)

[28] Fagin R, Lotem A, Naor M. Optimal aggregation algorithms for middleware[J]. Computer and System Sciences, 2003, 66(4): 614-656

[29] Sarawagi S, Kirpal A. Efficient set joins on similarity predicates [C]// Proc of SIGMOD. New York: ACM, 2004: 743-754

[30] Xiao C, Wang W, Lin X M, et al. Efficient similarity joins for near duplicate detection [C]// Proc of WWW. New York: ACM, 2008: 131-140

[31] Jacox E H, Samet H. Metric space similarity joins [J]. ACM Transactions on Database Systems, 2008, 33(2): 1-38

[32] Xiao C, Wang W, Lin X M, et al. Top-k set similarity joins[C]// Proc of ICDE. Washington: IEEE, 2009: 916-927

[33] Zhu S W, Wu J J, Xia G P. Top-k cosine similarity interesting pairs search [C]// Proc of FSKD. Washington: IEEE, 2010: 1479-1483

[34] Blanas S, Patel J M, Ercegovac V, et al. A comparison of join algorithms for log processing in MaPreduce[C]// Proc of SIGMOD. New York: ACM, 2010: 975-986

[35] Li G L, Deng D, Wang J N, et al. Pass-Join: a partition-based method for similarity joins[C]// Proc of VLDB. New York: ACM, 2012: 253-264

[36] Lynden S J, Tanimura Y, Kojima I, et al. Dynamic data redistribution for MapReduce joins[C]// Proc of Cloud Com. Washington: IEEE, 2011: 717-723

[37] Afrati F N, Ullman J D. Optimizing multiway joins in a Map-Reduce environment [J]. TKDE, 2011, 23(9): 1282-1298

[38] Zhang X F, Chen L, Wang M. Towards Efficient join processing over large rdf graph using MapReduce[C]// Proc of SSDBM. Berlin: Springer, 2012: 250-259

[39] Lin J. Brute force and indexed approaches to pairwise document similarity comparisons with MapReduce[C]// Proc of SIGIR. New York: ACM, 2009: 155-162

[40] Elsayed T, Lin J J, Oard D W. Pairwise document similarity in large collections with MapReduce[C]// Proc of ACL. Stroudsburg: Association for Computational Linguistics, 2008: 265-268

[41] Li B, Mazur E, Diao Y L, et al. A platform for scalable one-pass analytics using MapReduce[C]// Proc of SIGMOD. New York: ACM, 2011: 985-996

[42] Gufler B, Augsten N, Reiser A, et al. Handling data skew in MapReduce[C]// Proc of CLOSER. SciTePress, 2011: 574-583

[43] Gufler B, Augsten N, Reiser A, et al. Load balancing in MapReduce based on scalable cardinality estimates[C]// Proc of ICDE. Washington: IEEE, 2012: 522-533