外包空间数据库中的反向 k 最远邻居查询验证技术

王海霞 谷 峪 于 戈
(东北大学计算机科学与工程学院 沈阳 110169)

摘 要 由于数据爆发增长,数据拥有者不能高效处理客户端发送的查询请求,因此将数据外包给第三方数据发布者,委托第三方数据发布者来管理数据并且执行用户查询.当第三方数据发布者受到黑客攻击或者由于自身计算错误等情况发生时,将导致用户获取错误的查询结果.为了确保用户获得正确、完整且有效的外包空间数据库查询结果,查询验证技术得到了深入研究.此外,反向 k 最远邻居查询在近年来获得广泛关注.反向 k 最远邻居查询具有广泛的实际应用.例如,化工厂选址和基于位置的多人角色扮演游戏(如 BotFighters).在许多应用中,获得完全正确的查询结果是必要的.如果投建化工厂位置不合适,将会干扰居民和破坏环境.因此,有效和高效的反向 k 最远邻居查询验证技术对外包数据库是十分有价值的.该文基于已有的反向 k 最远邻居查询方法和 MR-tree 验证数据结构,首次提出了两种验证方法:一是 IZ-Auth 方法,将反向 k 最远邻居查询验证分解成反向 k 最远邻居范围验证和该范围内结果的验证两部分.该方法的客户端验证的首要任务是重塑根摘要,判断验证对象是否被篡改或者丢失,然后利用相关定理检验由半空间修剪技术形成的范围,只有完整的范围才能筛选出有效、正确且完整的反向 k 最远邻居查询结果.二是 UC-Auth 方法,先重塑根摘要来确保数据来源的可靠性,然后利用外围圆的特性检验验证对象和查询结果.UC-Auth 方法的优势在于其不需要计算 IZ-Auth 方法的范围,这能降低服务器端的计算开销.这两种验证方法是通过优化验证对象数量来降低通信和客户端验证代价.该文利用真实数据集和合成数据集进行了大量的实验,证明了这两种验证算法的有效性和实用性.该文提出的这两种验证算法可以将验证对象缩减至原始数据的 5%左右,既降低了通信代价,又提升了客户端验证效率.

关键词 外包空间数据库;反向 k 最远邻居查询;半空间修剪技术;验证数据结构;验证对象
中图法分类号 TP311 **DOI 号** 10.11897/SP.J.1016.2018.01896

Authentication Techniques of Reverse k Furthest Neighbor Queries in Outsourcing Spatial Databases

WANG Hai-Xia GU Yu YU Ge
(*School of Computer Science and Engineering, Northeastern University, Shenyang* 110169)

Abstract As the amount of data explodes rapidly, the data owner cannot handle queries efficiently that clients send. Therefore, the data owner outsources databases to a third-party data publisher and a third-party data publisher is delegated to manage the data and perform user queries. When a third-party data publisher is attacked by a hacker or due to own fallacious calculation, incorrect results will be returned to the user. In order to ensure that users get the correct, complete and valid query results in outsourcing spatial databases, the authentication techniques of queries have been widely explored. Additionally, in recent years, the reverse k furthest neighbor query has caused widespread attention. Given a set of facilities F, a set of users U and a query object $q \in F$, the reverse k furthest neighbor query retrieves each user $u \in U$, if q becomes one of the u's furthest k

收稿日期:2017-10-10;在线出版日期:2018-04-08.本课题得到国家自然科学基金(61472071,61433008)、中央高校基本科研业务费专项资金(N171605001)、辽宁省自然科学基金(2015020018)资助.**王海霞**,女,1991 年生,硕士研究生,主要研究方向为时空数据管理.**谷 峪**(通信作者),男,1981 年生,博士,教授,中国计算机学会(CCF)高级会员,主要研究领域为时空数据管理、图数据管理、大数据分析等.E-mail:guyu@mail.neu.edu.cn.**于 戈**,男,1962 年生,博士,教授,中国计算机学会(CCF)会士,主要研究领域为分布式和并行数据库管理、OLAP 和数据仓库、数据集成、图数据管理等.

facility objects. The reverse k furthest neighbor query can support many practical applications, such as location selection of building chemical factory, and location-based multiplayer role-playing games (such as BotFighters) among others. Specifically, totally correct query results are necessary in a lot of scenarios. For example, if the location of building chemical factory is improper, which will interfere with the life of residents and damage the environment. Therefore, the effective and efficient authentication technique of the reverse k furthest neighbor query is valuable in outsources databases. Based on the existing reverse k furthest neighbor query methods and the authentication data structure of MR-tree, this paper firstly presents two feasible authentication methods of the reverse k furthest neighbor query. The first method (IZ-Auth method) is broken down into two parts: the range verification of the reverse k furthest neighbors query and the validation of the results within that range. The primary task of IZ-Auth in client authentication is to reconstruct the digest of the root node in MR-tree, which determines whether verification objects have been tampered or lost, and then relevant theorems are proposed to verify the integrity of the range that is generated by the half-space pruning techniques. Only the complete range can filter out the valid, complete and correct query results. The second method (UC-Auth method) uses the characteristic of the outer circle to verify the correctness, validity and integrity of verification objects and the reverse k furthest neighbor query results, after reconstructing the digest of root node in MR-tree to ensure the reliability of the data source. Specifically, UC-Auth method does not need to calculate the range involved in IZ-Auth method, which can reduce the computational overhead on the server side. By reducing the amount of verification objects, the proposed two authentication methods can improve the communication and client verification cost. The adaptability of the two methods is also discussed in the paper. Finally, the effectiveness and efficiency have been verified by extensive experiments using real data sets and synthetic data sets. The proposed two authentication algorithms can reduce the verification objects to about 5% of the original data set, which not only reduce the communication cost, but also improve the client verification efficiency.

Keywords　outsourcing spatial database; reverse k furthest neighbors; half-space pruning techniques; authentication data structure; verification objects

1　引　言

由于数据爆发式增长，数据拥有者受到存储和计算能力的限制，不能高效地处理用户请求，于是将数据委托给有能力的第三方数据发布者来管理和执行用户查询，这使得空间数据库外包成为一种必然趋势. 然而第三方返回的查询结果可能被篡改或者丢失，导致用户获得错误结果，所以客户端需要验证结果，从而决定是否反馈给用户.

近年来，外包空间数据库的查询验证引起了研究者和社会的广泛关注. 一些学者提出了：范围查询验证[1]、k 近邻查询验证[2]、skyline 查询验证[3-4]、移动 k 近邻查询验证[5-6]、基于路网的 k 近邻查询验证[7-8]以及反向 k 近邻查询验证[9-10]等，其中范围查询验证和 k 近邻查询验证的验证技术相对成熟.

反向 k 最远邻居查询[11-15](Reverse k Furthest Neighbors，RkFN)近年来得到了广泛的关注. 给定 facilities(设施对象)集合 F，users 集合 U 和查询对象 $q\in F$，RkFN 查询检索每个用户 $u\in U$，若 q 成为 u 的最远 k 个设施对象之一，那么返回用户 u. RkFN 查询在实际中有广泛应用. 例如，生活中化工厂等危害设施的投建选址问题，应该使得更多用户受到最小影响，这需要 RkFN 查询方法作为查找手段，在候选的设施位置中选取 RkFN 结果数量最多(受到最小影响的用户最多)的位置. 再如，在基于位置的多人角色扮演游戏(如 BotFighters)中，其他玩家往往关注近邻玩家射击而忽视距离远的玩家，因此 RkFN 查询可以定位那些对查询玩家防范度最低的玩家，射击这些玩家可能会有更好的效果，同样的策略也可在真实的战场中应用[11]. 在以上的例子当中，查询结果正确是非常重要的，否则可能造成严

重的后果.这就需要通过 RkFN 查询验证技术对获取的结果进行检验,确保查询结果的正确性.

最近,作者 Wang 等人[11]提出了一种新的 RkFN 查询算法,相比前人提出的反向最远邻居(Reverse Furthest Neighbor,RFN)算法[12-15]进行改进,扩展了 $k=1$ 的限制,能高效地处理用户请求.本文基于海量数据和外包风险的背景,针对 RkFN 查询方法[11]进行了深入研究,率先提出了两种优化方法 IZ-Auth 和 UC-Auth,旨在分别提高客户端和服务器端的处理效率,从而满足不同场景下的客户端校验,继而确保用户获取正确的查询结果.

IZ-Auth 方法和 UC-Auth 方法的思想是优化大规模的验证对象(Verification Object,VO),继而降低通信和客户端验证代价.IZ-Auth 方法的客户端验证算法分为三个阶段:第一阶段采用重塑根摘要的策略来检验 VO 是否可靠,即是否出现被篡改或者丢失等错误;第二阶段利用相关定理验证由 shallow facility[11]形成的 IZ,该 IZ 的正确性保障了第三阶段验证的可信性;第三阶段按照范围查询验证的思想,利用正确的 IZ 区域来筛选出满足 RkFN 查询条件的结果,进而判断出结果的正确性、有效性和完整性.这三个阶段环环相扣,从而保证了 RkFN 查询验证的正确性.UC-Auth 方法的客户端验证算法需要两个阶段实现.第一阶段同 IZ-Auth 方法的第一阶段;第二阶段利用外围圆的特性来检验 VO 和 RkFN 查询结果的正确性.IZ-Auth 方法具有较高的验证效率,能充分降低通信开销.UC-Auth 方法并不需要计算 IZ 边界区域,这直接降低了服务器端的计算开销和传输 IZ 的通信代价,然而却可能增加客户端验证查询结果的花销.

本文主要有以下贡献点:

(1)最先提出了反向 k 最远邻居查询验证的方法,该方法适用于 RkFN 查询结果的验证;

(2)设计了两种压缩 VO 的方法,继而降低了通信和验证等相关代价;

(3)针对影响区域(IZ)和外围圆的验证方法进行了严密地推导,保证了查询验证的正确性;

(4)本文提出的两种验证方法经过了大量真实和合成数据的检验,实验证明了验证算法的有效性和实用性.

本文第 2 节是相关工作概述;第 3 节是 RkFN 查询验证技术的问题描述;第 4 节是 RkFN 查询验证方法;第 5 节是实验评估;第 6 节是实验总结;最后是参考文献.

2 相关工作

2.1 外包数据库

外包数据库[16-17]是网络和大数据时代应运而生的产物,数据拥有者由于存储能力和计算能力的限制,导致不能为大规模的用户请求进行高效地处理,所以将数据库外包给第三方数据发布者.外包数据库必然会带来风险,数据的可靠性受到质疑,即第三方将处理结果返回给用户,用户会质疑该结果是否存在丢失、被篡改等情况.为了消除用户的顾虑,查询验证会把相关的 VO 和根签名(Root Signature)附加给用户,以便用户检验查询结果.图 1 是数据库外包框架,数据拥有者(Data Owner)将数据集、Root Signature、验证数据结构(ADS)发送给第三方数据发布者,第三方根据 ADS 和 Root Signature 来检验数据集的正确性.同理,第三方数据发布者将处理的查询结果同 VO、Root Signature 发送给客户端,利于客户端验证查询结果正确性.

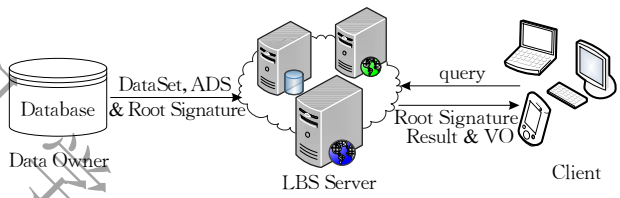


图 1 数据库外包框架

目前,ADS 有两种验证结构,即 MR-tree[16]和 Signature Aggregation and Chaining[17],但是验证高维空间数据库的查询,一般会采用 Yang 等人[16]提出的 MR-tree 验证索引树.在 MR-tree 中,每个节点都有其摘要信息(使用一种单向的 $hash$ 函数来计算摘要),摘要信息分为两种形式:叶子节点摘要和中间节点摘要.不管是第三方获取数据拥有者的数据,还是客户端接受第三方返回的查询结果,只要确保解密后的根节点摘要信息正确,就能证明数据来源的可靠性.

2.2 空间查询验证

范围查询验证[1]是基于 MR-tree 验证索引树检索出该范围内的对象返回给用户,然后利用深度遍历算法有序地加载出 VO,利用该 VO 验证结果的正确性.skyline 查询验证[4]采用 Partial-S4-tree 比 MSR-tree 能更好地处理大规模数据和子图空间,可以有效地过滤掉验证对象集合中大量的冗余对象,进而提高验证效率.移动 k 近邻查询验证[5-6]实质上是安全区域的验证,利用影响对象来验证安

全区域的完整性、正确性. 如果 q 在安全区域内移动，则查询结果不变，一旦 q 离开安全区域，则用户需要提出新的查询请求. 设计安全区域的目的是降低客户端和服务器的交互频率，减缓服务器的计算压力. 基于路网的 k 近邻查询验证[7]采用基于路网 Voronoi Diagram 的方式来验证结果. 反向 k 近邻查询验证[9-10]利用包围圆来形成验证对象，借助验证对象来检验结果的正确性、有效性和完整性.

2.3　反向 k 最远邻居查询

RFN 查询[12-15]是 k 为 1 的 RkFN 查询. Yao 等人[12]提出了 Convex Hull Furthest Cell 算法来计算 furthest voronoi cell，该算法是基于 R-tree 来处理数据集的 convex hull 和 Voronoi diagrams，进而计算出 RFN 查询结果. 然而，在 R-tree 上计算 convex hull 和执行范围查询相当耗时. Liu 等人[13]提出了 PIV 算法来构建 RFN 索引结构并利用三角形不等式进行修剪处理，该索引结构采用了 convex hull 特性提高查询效率，避免了 Yao 等人[12]提出的基于 R-tree 动态计算代价. Xu 等人[14]采用 landmarks 和 HP-tree 来修剪离查询点并不远的数据点，这样比 BFS 方式要少访问 3%个数据点. Li 等人[15]针对移动对象的反向最远邻居的查询处理，以 F-BFS 方式来遍历 TPR-tree，过滤掉不包含 DRFN 的带时间参数的外包矩形框或者移动数据对象，这样能缩减搜索空间，提高查询效率. 以上都是针对 RFN 的静态或者动态的查询处理现状. 最近，Wang 等人[11]提出了 RkFN 查询方法，突破了 k 为 1 的限制，使其更具有实际效用. RkFN 查询思想是利用 k-depth contour[18]对空间对象进行约束，若查询对象 q 处于 k-depth contour 内部，则 q 是一个无效查询，k-depth contour 是有效与无效查询的分界线，能高效地筛选出有效查询. 图 2 中 q 与对象 f_1 至 f_8 形成的半空间中包含的对象数目满足 $|f_i|\leq 2$ 的补集空间是 2-depth contour，即实线围成的多边形. 本文采用 k-depth contour 的理念来验证 RkFN 查询，这样能缩减 VO，降低通信和计算等代价.

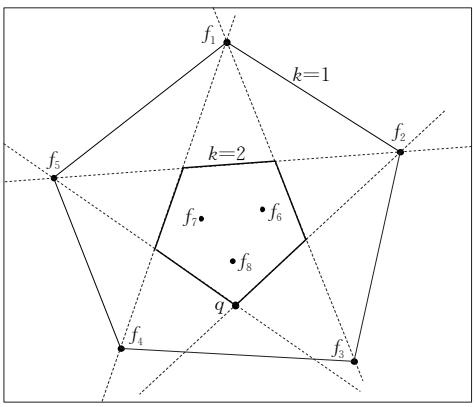


图 2　k-depth contour($k=2$)

3　问题描述

本节给出 RkFN 查询验证所涉及到的基本概念，包括 RkFN 查询验证需要满足的条件. 表 1 是本文经常使用的符号及其含义.

表 1　基本符号表示

名称	描述
q	查询点
F	设施(Facilities)对象的集合
U	用户(Users)对象的集合
fVO	F 的验证对象
S_F	F 的空间边界
uVO	U 的验证对象
IZ	q 成为某些对象的最远 k 个邻居之一的区域
$dist(p,q)$	表示点 p 到点 q 的欧式距离
$\Box(q,p)$	圆心为 q，半径为 $dist(q,p)$ 的圆外区域

3.1　问题定义

定义 1.　影响区域(IZ). 判断 IZ 内任意一点 p 到查询点 $q\in F$ 形成的包围圆外部是否包含不少于 k 个设施对象，如果满足，则返回 false，否则，返回 true. 其形式化表示为式(1).

$$Range\text{-}k(q,k,F,IZ)=\{p\in IZ \mid count(F\bigcap \Box(p,q))<k\} \quad (1)$$

定义 2.　Location depth(位深)[11]. 查询点 $q\in F$ 的位深用 $|q|$ 表示，是所有通过 q 的直线 L_q 形成的半空间的 $depth$(深度) H_q 的最小值，即为位深 $|q|$.

定义 3.　k-depth contour 是一个凸多边形. 对于任意一点 p 而言，需要满足以下条件：(1) 若点 p 存在该 k-depth contour 上或者内部，则 $|p|\geq k$；(2) 若点 p 严格存在于 k-depth contour 外部，则 $|p|<k$.

3.2　查询验证满足的性质

第三方把根签名、VO 和结果发送给用户，以便客户端通过该 VO 检验结果是否满足查询要求，为了验证 RkFN 查询结果的正确性，其查询验证需要满足以下 3 个性质：

(1) 正确性. 保证用户从第三方服务器获得的查询结果在原始数据服务器中存在，并且和原始数据服务器中的数据保持一致.

(2) 有效性. 保证用户获得的结果信息满足用户查询请求，对于反向 k 最远邻居查询验证，结果应

满足 RkFN 的查询条件.

(3)完整性. 保证数据没有遗漏或者被篡改,继而满足查询条件的结果点.

由此可见, RkFN 查询验证就是客户端检验结果是否满足上述三个性质,一旦发现结果不满足任何一个性质,则说明客户端获取了错误的结果,可直接丢掉第三方返回的结果.

IZ-Auth 和 UC-Auth 方法都需要利用验证对象来重塑根摘要,判断数据来源的正确性. IZ-Auth 方法之后会利用 fVO 检验 IZ 的完整性,继而利用正确的 IZ 筛选 uVO 中满足查询条件的对象,以此对比第三方返回的查询结果的三性是否正确. UC-Auth 方法是通过 fVO 和 uVO 互相检验的机制来判断结果的正确性,该互检机制采用了外围圆的特性来验证结果三性的正确性.

本文的重点是客户端如何验证出不完整的 IZ (注:本文所有实例中 k 的取值均为 2). 如图 3 所示,q 的 RkFN 结果为 $C_1 \sim C_9$ 围成的封闭区域所包含的 user 对象. 图 3 出现了三种 IZ 不完整的情况:缺失 ΔPC_1C_2、$\Delta C_2C_3C_4$、增多 $\Delta C_1C_2C_3$. 本文提出的 IZ-Auth 方法不同于 UC-Auth 方法之处在于将 RkFN 查询验证分解为 VO 和 IZ 两部分,借助 IZ 来有效地验证结果. 而 UC-Auth 方法利用外围圆特性来保证只利用验证对象就能达到验证结果的目的,省去了服务器端额外计算 IZ 的开销,具体验证过程将在第 4 节进行说明.

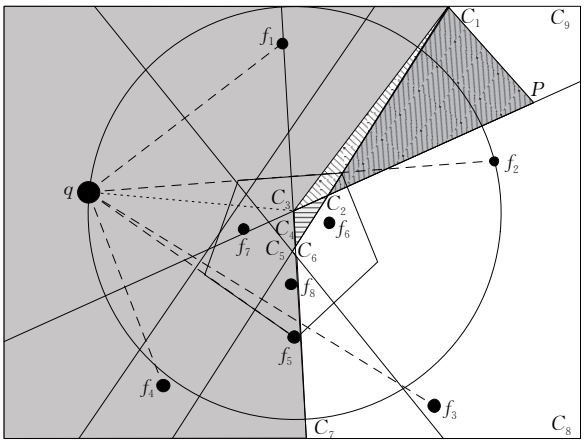


图 3 R2FN 的影响区域

4 RkFN 查询验证算法

本节分为两个环节,一是第三方服务器端生成 RkFN 结果及其验证信息;二是客户端利用验证信息检验 RkFN 结果. 第二环节可采用本文提出的任意一种验证方法来验证. IZ-Auth 和 UC-Auth 方法都要先通过 fVO 和 uVO 来判断数据来源的可靠性. 之后,IZ-Auth 方法利用 fVO 验证 IZ 区域的完整性. 最后在 uVO 中进行 IZ 范围查询,继而判断结果三性是否正确. UC-Auth 方法经过可靠性验证之后,需要借助外围圆的检验机制来验证查询结果. 若客户端验证处理的任何一个环节出错,则算法终止,即判定用户获取了错误的 RkFN 查询结果.

算法 1 是服务器采用 IZ-Auth 方法生成 RkFN 结果及其验证信息的过程. 第 1 行可以通过文献[18]来生成 k-depth contour,这可以根据实际查询情况离线生成,也可在线生成;第 2～14 行是 fVO 的生成过程,该过程依托于 k-depth contour,亦可以离线生成;第 15～25 行是 IZ 边界区域的形成,其中第 15 行是利用 fVO 内数据点来形成修剪区域 IZ;第 26～28 行产生了 $result$ 和 uVO,该过程可以并行执行,比如生成 $result$,可以将数据集分割,并行执行每部分数据,采用射线法判断对象是否在 IZ 内部,以此发挥云服务器端的性能. 最后第三方服务器将 IZ、$result$、fVO 和 uVO 发送给客户端,供客户端检验结果. 若服务器采用 UC-Auth 方法生成 RkFN 结果及其验证信息,其思想同 IZ-Auth 方法的生成过程,只是 UC-Auth 方法省略了生成 IZ 边界的步骤.

4.1 服务器端算法

算法 1. 生成 IZ, fVO, uVO, $result$.

输入:facilities 索引树 F-MR-TREE; users 索引树 U-MR-TREE;查询设备 q;用户请求参数 k

输出:客户端验证 RkFN 查询结果的验证信息,包括 IZ, fVO, uVO;服务器端生成查询结果 $result$

```
1.  K_v ← Generate the k-depth contour of F
2.  Initialization IZ by S_F and s_root ← the root of F
3.  WHILE (s_root)
4.     Dequeue(s_root) //进行深度遍历获得 fVO 和 IZ
5.     IF s_root ⊂ K_v
6.        fVO ← s_root
7.     ENDIF
8.     ELSE Queue(s_root.childs)
9.        IF s_root.childs are leafs in F-MR-TREE
10.          fVO ← s_root.childs
11.          queuePruners(s_root.childs)
12.       ENDIF
13.    ENDELSE
14. ENDWHILE
15. WHILE(queuePruners(s_root.childs))
16.    f_i ← Dequeue(queuePruners(s_root.childs))
17.    B_{q:f_i}
```

18. FOR label zones P^j_{zone} that were prune
19. $count_j \leftarrow$ update P^j_{zone}
20. IF $count_j \leq k$
21. delete P^j_{zone}
22. $IZ \leftarrow$ update(delete P^j_{zone})
23. ENDIF
24. ENDFOR
25. ENDWHILE
26. Find the *result* in U-MR-TREE that $u_i \in U$ is in IZ
27. Compute uVO in U-MR-TREE by IZ
28. Send IZ, fVO, uVO, *result* to the Client

4.2 客户端验证算法

根据 MR-tree 结构，将重构的根摘要与解密后的根摘要对比. 若两者一致，则说明数据来源可靠. 继而利用 VO 和 IZ 来验证结果的三性. 若验证出 IZ 不完整，则说明可能会丢失部分满足查询条件的结果或者增加部分不满足用户查询请求的结果.

情况 1. IZ 存在.

定义 4. IZ 顶点及 IZ 内任意一点 p 到 q 形成的¤(p,q)与 fVO 交集的对象数目小于 k，即形式化表示为式(2).

$$count(¤(p,q) \bigcap fVO) < k \tag{2}$$

定理 1. IZ 边是由 q 和 fVO 中某些对象的中垂线构成，或者可能外加部分数据空间 S_F 边界线构成.

证明. 显然地，非 S_F 构成的 IZ 边都是由 shallow facility[11] 和 q 的中垂线构成，该对象必在 fVO 中. 假设某条边由非 fVO 中数据点和 q 的中垂线构成，那么 shallow facility 会把该粗粒度的边进行精确，故不存在该边. 证毕.

定理 2. IZ 边 e_i 上非 IZ 顶点的任一点 v_i 到 q 形成¤(v_i,q)与 fVO 交集所包含的对象数目等于 $k-1$，若存在由 S_F 顶点组成的边 e_s，该边上除了 IZ 顶点外的任一点 v_s 到 q 形成¤(v_s,q)与 fVO 交集所包含的对象数目不超过 k-1.

无数据空间顶点组成的边形式化表示为式(3).

$$\begin{aligned} &count(¤(e_i,q) \bigcap d_k) = k-1 \\ &count(¤(e_i,q) \bigcap M_k) = 0 \end{aligned} \tag{3}$$

有数据空间顶点组成的边形式化表示为式(4).

$$\begin{aligned} &count(¤(e_s,q) \bigcap d_k) < k-1 \\ &count(¤(e_s,q) \bigcap M_k) = 0 \end{aligned} \tag{4}$$

注：d_k 表示的是 fVO 中数据对象，而 M_k 代表的是 fVO 中矩形实体对象.

证明. 由定义 4 可知，IZ 顶点 V 是由至少 k 个设施对象分别与 q 做 $B_{q:f}$ 而形成的，故必满足 $count(¤(V,q) \bigcap fVO) \leq k-1$，才能使得 q 成为 k-th 设施对象. 无数据空间顶点组成的边是由 k 个设施对象和 q 做 $B_{q:f}$ 形成的，那么边上任意一点 p 到 q 形成的¤(p,q)与 fVO 相交的对象数目均为 $k-1$，而由数据空间顶点组成的 IZ 边至多有 k 个对象与 q 做 $B_{q:f}$ 形成该边，故 IZ 边上任意一点到 q 形成的外围圆包含的对象数目必定小于 k. 证毕.

定理 3. IZ 内部的所有数据点 p 到 q 所形成的¤(p,q)都不相交于 fVO 中矩形实体.

证明. 假设 IZ 内部存在点 v_i，且¤(v_i,q)与矩形实体相交，那么 k-depth contour 上至少存在一个顶点 v_j，$dist(v_j,v_i) > dist(q,v_i)$，且 k-depth contour 上各个顶点的位深 $|v_i| \geq k$[11]，每个顶点都由至少 k 个对象修剪而成，这就形成了 $B_{v_i:q}$ 半平面，这与原始设计 $B_{q:v}$ 相矛盾，故该定理成立. 证毕.

客户端针对非空 IZ 的验证过程如下. 首先，检验 IZ 内部的所有数据点 p 到 q 形成的¤(p,q)是否都不相交于 fVO 中矩形实体，若相交，则验证失败. 其次，验证 IZ 顶点是否满足定义 4. 如果某顶点 p 的¤(p,q)包含的对象数目不少于 k，则算法终止. 然后，利用定理 1 来验证 IZ 边的构成成分是否正确，一旦发现错误，验证终止，否则采用定理 2 来进行最后 IZ 完整性验证. 如图 3，若 IZ 多出一块 $\Delta C_1C_2C_3$ 区域，则可用定理 1 来判定. 客户端检测出 q 通过边 C_1C_3 找不到对称点 f，且该边也不是数据空间边界，因此，验证出 IZ 区域不完整. 若 IZ 缺失 $\Delta C_2C_3C_4$ 区域，则可用定理 2 来检验，定理 2 采用计数修剪法[21-22]来验证，先找到 q 关于边 C_1C_5 对称的对象 f_3，并累计端点 C_1、C_5 被修剪的次数；接着查找是否还有其它对象能修剪该边，找到 f_4，而 f_4 修剪 C_2C_1，新增节点 C_2，继续累计新旧节点被修剪的次数；按照该方式依次类推，最后求得 $count(C_1)=2$，$count(C_5)=2$，$count(C_2)=1$，$count(C_4)=1$，判断 C_1 到 C_5 是否有被修剪次数小于 k，即 $count(C_i)<k$ 且该边不包含 S_F 边界点，其中 C_1C_5 上的点的 $count$ 值不同，说明该边不正确，这如同检查电路故障，查看电流是否相等，电线粗细是否均匀等问题.

情况 2. IZ 不存在.

如果 IZ 不存在，则不能利用 IZ 的顶点与边的相关定理来验证结果为空的情况.

定理 4. 如果每个象限有至少 k 个对象，那么查询点 q 没有任何 RkFN.

证明. 如图 4 所示，毋容置疑，我们证明 Q_3 象限没有 RkFN. 考虑 Q_3 象限的点 p 和 Q_1 象限的点 f，令 p_x，p_y 是 p 的 x 和 y 坐标，对于 Q_1 象限中任何

一点 f,我们有 $|p_x-f_x|>|p_x-q_x|$ 且 $|p_y-f_y|>|p_y-q_y|$,因此,$dist(p,f)>dist(p,q)$,如果 Q_1 象限至少有 k 个对象,q 将并不会成为 Q_3 象限中 p 点的 RkFN. 证毕.

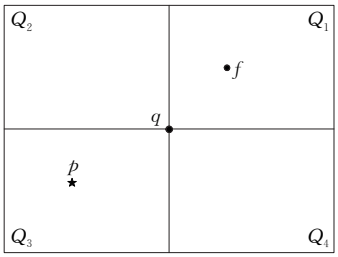


图 4 四象限测试

IZ 为空有两种情况:一是查询点 q 在 k-depth contour 内部;二是查询点 q 在 k-depth contour 上或者外部,仍使得 IZ 为空.

针对第一种处于 k-depth contour 内部的无效查询的情况,其形成的 IZ 为空. 如图 5,q 严格存在 k-depth contour 内. 我们采用四象限测试法验证结果为空,以 q 点划分出四个象限,统计各象限包含 fVO 数据对象数目. 若某象限的对象数目不小于 k,则对角线的象限区域被修剪. 由于 C 象限中有 $\{f_3,f_5\}$,D 象限中有 $\{f_1,f_2\}$,故 A 和 B 象限被修剪,接着通过 A 和 B 象限中 $\{f_4,f_7\}$ 来修剪 C 和 D 象限,致使 D 象限被修剪,C 象限最后被 D 象限 $\{f_2\}$ 修剪掉,致使 q 没有 RkFN.

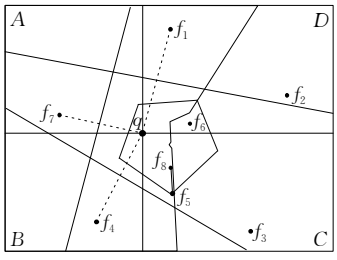


图 5 fVO 验证 IZ 为空

而第二种处于 k-depth contour 上或者外部的查询 q,仍然会形成空 IZ. 那么这种情况的检验方式同第一种情况,只不过该情况的验证比第一种的计算量稍大,但整体验证并不耗费时间.

算法 2. 客户端采用 IZ 验证 RkFN 查询结果.

输入:第三方服务器返回的查询结果 *result*;服务器生成的影响区域 IZ;服务器产生的验证对象 fVO,uVO;用户查询请求 q;用户请求参数 k

输出:RkFN 查询结果的正确性 true/false

```
1.  IF IZ=∅ AND result!=∅
2.      RETURN false;      //IZ 和 result 相互矛盾
3.  ELSE h'_f=reconstruct the root digest from fVO
         h'_u=reconstruct the root digest from uVO
4.  IF h'_f!=h^f_root || h'_u!=h^u_root
5.      RETURN false;
6.  IF IZ=∅
7.      USING Theorem 4 to verify the result
8.  IF IZ!=∅
9.      FOR each v_i∈V
10.         IF count (¤(v_i,q)∩fVO )≥k
11.             RETURN false;      //定义 4
12.     FOR e_i=v_iv_{i+1}; v_i,v_{i+1}∈V
13.         IF e_i∉S_F
14.             IF Theorem 1 is false
15.                 RETURN false;
16.             ELSE IF count(¤(e_i,q)∩fVO)≠k-1
17.                 RETURN false;  //定理 1 和 2
18.         IF e_i∈S_F && count(¤(e_i,q)∩fVO)≥k-1
19.             RETURN false;
20. FOR each object p_i of uVO
21.     IF p_i∈IZ
22.         result'←p_i
23. IF result'==result
24.     RETURN true;
25. Verify the result is correct, receive the result
```

算法 2 是客户端采用 IZ 思想来验证 RkFN 查询结果的正确性,其验证过程与范围查询验证类似. 其中第 3~5 行是客户端通过 fVO 和 uVO 来重塑根摘要,确保数据来源可靠. 第 6、7 行采用定理 4 来验证 IZ 为空的情况;第 8~19 行针对非空 IZ 情况,客户端利用 fVO 来检验 IZ 的完整性. 最后,第 20~25 行说明了如何用 uVO 和 IZ 来验证 *result* 三性,其中 IZ 的作用是保证 RkFN 查询结果的完整性.

4.3 优化 IZ 验证算法

上一部分在验证 IZ 边时,客户端采用了计数修剪法[19-20]. 通过 k-depth contour 外部的对象与 q 形成的中垂线来修剪 IZ 的每条边,判断每条边被修剪的次数是否满足 RkFN 条件,然而这种验证 IZ 边的代价对于客户端而言比较高昂,本文提出了向量探测法来验证 IZ 各边情况.

定义 5. 探测点/内点/外点. 给定 IZ 中任意一个顶点 v_{cur},在形成 v_{cur} 的中垂线两侧分别取靠近 v_{cur} 的任意点 p_1 和 p_2,该点 p_1 和 p_2 点称之为探测点. 如果探测点处于 IZ 区域内部,则称之为内点;反之,处于 IZ 区域外部的探测点,称之为外点.

性质 1. 内点/外点性质. 给定任意内点 v_{in} 或

外点 v_{out}，则 v_{in}/v_{out} 应满足式(5).

$$count(\square(v_{in},q)\cap fVO)<k \qquad (5)$$
$$count(\square(v_{out},q)\cap fVO)\geq k$$

向量探测法是通过判断探测点是否满足内点和外点的性质，从而确定 IZ 边界的完整性的方法. 如图 6 所示，对于出现缺失 $\Delta C_2C_3C_4$ 这种情况，导致 C_1C_5 现在连成一条线段，C_5 此时的邻居是 C_1、C_6. 由于 fVO 中 f_4 与 q 的中垂线会相交于 IZ 边 C_1C_5，生成新节点 C_2，当客户端验证到 C_2 时，会沿着该节点所在的所有中垂线两侧取靠近该节点的任意探测点，即 C_2 所在的中垂线有 q 和 f_3 形成中垂线 $line_{C_1:C_5}$ 和 q 和 f_4 形成中垂线 $line_{C_2:C_3}$. 在中垂线 $line_{C_1:C_5}$ 上取靠近 C_2 的两侧取探测点 A 和 D，在 $line_{C_2:C_3}$ 上取靠近 C_2 的两侧取探测点 B 和 C，然后判断探测点 A，B，C，D 是否满足性质 1. 这里 A，B，C，D 到 q 点形成的圆外区域包含的对象数目都小于 2，但是点 C 明显是外点，这说明了 IZ 边界不完整，则验证失败. 对于 IZ 中出现增多区域($\Delta C_1C_2C_3$)情况，首先检查 IZ 顶点是否满足定义 4；然后检验 IZ 边是否由 q 和 fVO 中某数据对象的中垂线构成；最后通过向量探测法验证 IZ 边是否正确. 向量探测法能加快验证速度，提高验证效率，不用记录每个区域块被修剪的次数，大大简化了计算过程，改进了计数修剪法复杂、效率低的特点.

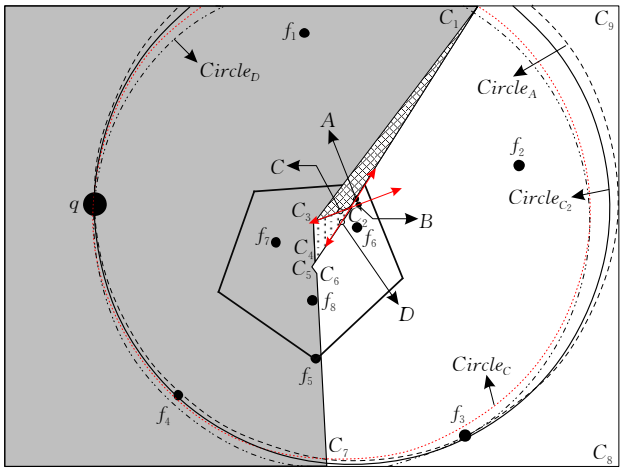


图 6　增多 $\Delta C_1C_2C_3$、缺失 $\Delta C_2C_3C_4$

用户选取查询验证技术的最根本目的是基于精准结果进行分析和预测，所以在保证查询结果正确、有效且完整的前提下，考虑优化时间性能. IZ-Auth 方法设计意义在于提高客户端的验证效率，让用户有良好的体验. 查询验证通常是利用验证对象来检验结果的三性是否正确，而 IZ-Auth 方法的创新之处在于辅助了 IZ 来检验结果的三性，正因为引入 IZ，可以直接提炼出结果，故省去了候选集验证的开销.

4.4　外围圆验证

定理 5.　点 p 和 q 是线段 pq 的两个端点，p' 是 pq 上任一点，那么 p' 的包围圆被 p 的包围圆包含.

定理 6.　给定一条线段 AB 和一个点 p，点 p 在 AB 上，那么点 p 的包围圆被线段 AB 两端点的包围圆的并集包含.

定理 5 和定理 6 的证明参考文献[19].

定理 7.　给定凸包多边形 $A_1A_2A_3\cdots$，凸包多边形内任一点的包围圆都被该多边形所有顶点包围圆的并集所包含.

证明.　如图 7，假设 p' 是凸包多边形内任一点，直线 $p'q$ 与凸包多边形一条边 AB 相交于点 p，由定理 4 可知，点 p' 的包围圆被点 p 的包围圆包含，由定理 5 可知，点 p 的包围圆被 p 所在边的两端点的包围圆的并集包含. 如果点 p 和它所在边的任意一点(A 或 B)重合，则由定理 4 可知，点 p' 的包围圆被点 A 或 B 的包围圆包含.　　证毕.

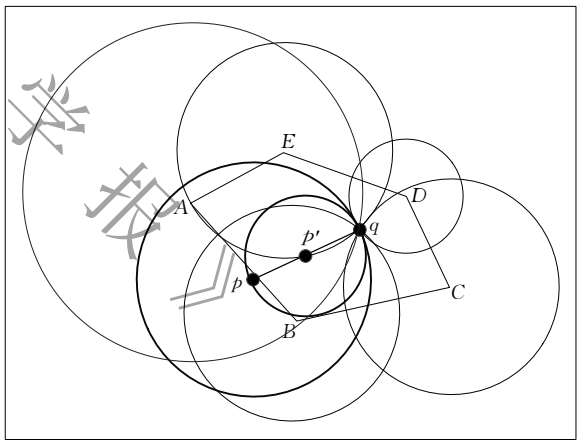


图 7　凸多边形的外围圆性质

定义 6.　矩形外围圆并集($\bigcup_{\square}^{M}$). 给定矩形 M 和查询点 q，则 $\bigcup_{\square}^{M}$ 表示矩形 M 的所有顶点 v^M 与 q 形成的 $\square(v^M,q)$ 的交集区域.

定理 8.　若矩形实体 M 满足式(6)，则矩形 M 处于正确状态；否则矩形 M 处于待检状态.

$$count(\bigcup_{\square}^{M}\cap fVO)\geq k \qquad (6)$$

证明.　因为矩形 M 的四个顶点的外围圆并集包含的 fVO 对象数都不小于 k，则矩形内任一点的外围圆包含的 fVO 对象的数目都不小于 k，故该矩形 M 一定不包含 RkFN 查询结果，因此该矩形 M 处于正确状态. 否则，矩形 M 处于待检测状态，需要

客户端采用其它手段进行检验. 证毕.

如图 8,矩形 $ABCD$ 四个顶点到 q 形成的外围圆并集包含至少 4 个对象 f_2,f_3,f_4和 f_5,故该矩形内的任何点到 q 形成的外围圆都满足 RkFN 查询条件. 所以,只要验证出矩形 M 的$\bigcup_{¤}^{M}$包含的对象数目不小于 k,就能证明矩形不包含 RkFN 查询结果. 如果矩形包含了 RkFN 查询结果,那么该矩形的$\bigcup_{¤}^{M}$交于 fVO 中数据对象的数目一定小于 k. 反过来,若该矩形的外围圆并集包含的对象数目小于 k,并不能说明该矩形错误.

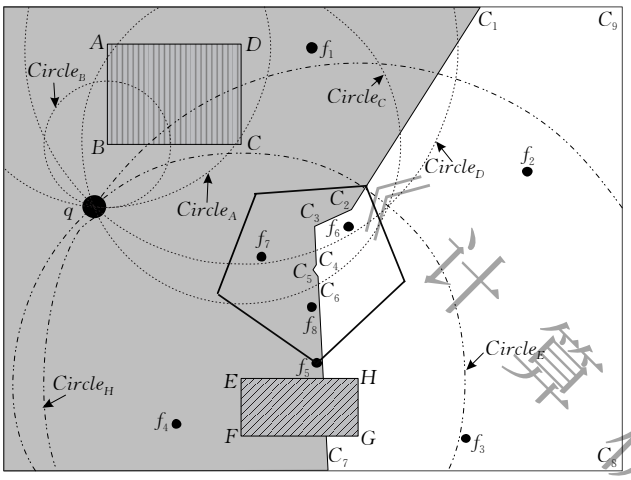


图 8 MBRs 外围圆并集包含的设施对象数超过 k

情况 1. $count(\bigcup_{¤}^{M} \cap fVO)<k$ 的错误情形. 若矩形某顶点的外围圆包含的对象数目小于 k,则说明该矩形包含结果. 如图 8,矩形顶点 H 和 G 到 q 形成的外围圆包含的对象数目小于 2,说明顶点 H 和 G 处于结果区域中,故该矩形可能包含查询结果.

情况 2. $count(\bigcup_{¤}^{M} \cap fVO)<k$ 的正确情形. 如图 9,矩形 $ABCD$ 显然正确,纵使其外围圆并集只包含 f_2,且小于 2. 所以,不能通过判断矩形外围圆并集包含的对象数目小于 k,就判定矩形错误,还需要通过其它手段来进一步验证该矩形的正确性.

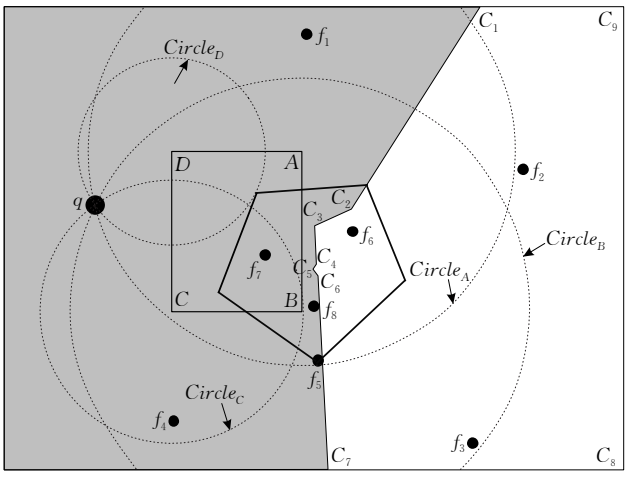


图 9 MBRs 外围圆并集包含的设施对象数小于 k

定义 7. 远点. 给定 fVO 中的任意一数据对象 p 和任意一矩形实体 M,如果 p 和 q 的中垂线相交于 M,则离 q 点较远的交点,称之为远点,记为 far_p.

定理 9. 给定远点 far_p,如果远点满足式(7),

$$¤(far_p,q)<k \tag{7}$$

则证明矩形 M 中包含查询结果. 否则,需要对 M 的包围圆并集内的所有对象进行 RkFN 查询检验,如果满足式(8),则表示矩形正确.

$$|\{o\in \bigcup_{⊙M} | ¤(o,q)\cap fVO\}|\geq k \tag{8}$$

通常,考虑离 q 较近的 fVO 中数据对象来验证 fVO 中的实体. 因为离 q 近的对象与 q 形成的中垂线交于实体的 far_p所形成的$¤(far_p,q)$将包含更少的对象,若及早发现 $¤(far_p,q)$包含的对象数目小于 k,则提前终止验证. 因此,采用远点检测法能解决无法判定的情况 1 和情况 2.

矩形外围圆并集与包围圆并集是互补关系,两者的并集为全集. 根据这个特点,我们提出了$|q|$位深的辅助远点检验的策略. 假设矩形 M 的$\bigcup_{¤}^{M}$包含的对象数为 $m(m<k)$,那么,在$\bigcup_{⊙M}$内查找使得$|q|$不大于 $k-m+1$ 的所有 fVO 数据对象,检查这些对象的远点的外围圆包含的对象数是否小于 k,因为位深值越小,可能暴露出的对象越多,这样能尽快地验证出错误的矩形. 如果满足位深条件的对象都不满足 RkFN 查询,则需要采用式(8)来检验矩形.

图 10 中 IZ 区域是检验结果正确性的边界线标志,只是便于直观判断结果的完整性,UC-Auth 方法中并不存在 IZ 区域. 由图 10 可知,矩形 $ABCD$

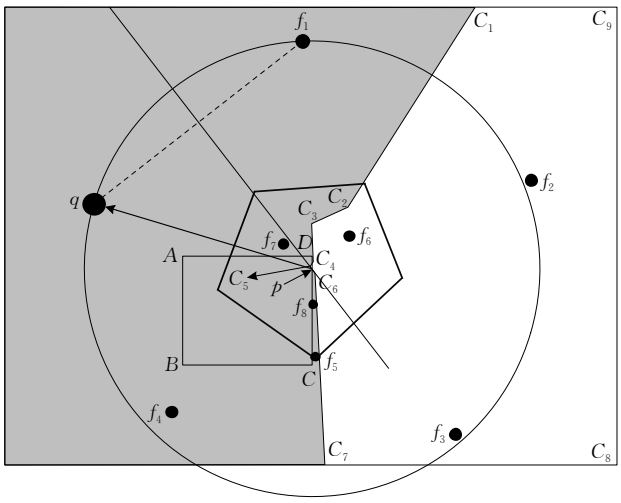


图 10 MBRs 外围圆并集包含的设施对象数小于 k

相交于 IZ 区域，则直观判定该矩形错误，而客户端通过计算才能判断出该矩形的状态. 首先，在矩形 M 的 $\bigcup_{\odot M}$ 内找到 $|q|$ 不超过 2 的所有对象，此时满足条件的对象有 f_1 和 f_4，其中 f_1 到 q 的中垂线交矩形 $ABCD$ 远点 p，而 $\square(p,q)$ 包含的对象数不超过 k，则说明 q 的 RkFN 查询结果不完整. 否则，需要验证矩形包围圆并集内的所有对象，判断其远点的外围圆包含的对象数是否超过 k，以此来判断该矩形的正确性，从而验证出 RkFN 查询结果的正确性.

UC-Auth 方法的创新性在于采用外围圆的性质来产生查询结果和验证对象，这种方式可以减少服务器端的计算开销，因为影响区域的计算开销为 km^2，随着数据量的增加，其计算开销较大. UC-Auth 方法可以让服务器减轻计算压力，为更多的用户提供高效的查询验证服务.

4.5　IZ-Auth 和 UC-Auth 的代价和适用性分析

最朴素的验证方法是将集合 F、集合 U 发送给客户端来校验结果，该方法简称为 BF-Auth. 验证方法的代价分析和实验评估中所使用的符号参看表 2. 假设集合 F 包含 n_1 个数据点，集合 U 包含了 n_2 个数据点，那么客户端采用 BF-Auth 方法验证 RkFN 查询结果的开销分为两部分. 第一部分是通过 fVO，uVO 重塑根摘要来检验数据来源的可靠性，其时间复杂度为 $O(n_1\log n_1+n_2\log n_2)$. 第二部分是验证 $result$ 正确性，其开销为 $n_1\times n_2$. 当第三方服务器采用该方法时，则其计算代价和验证代价相似，且其传输量需要包含 n_1+n_2 个数据点，客户端必须提供可以容纳 n_1+n_2 个数据的存储空间，才能进行后期验证操作，而客户端的存储和计算能力往往成为校验结果的瓶颈，因此本文提出了缩减验证对象的 IZ-Auth 和 UC-Auth 验证方法.

表 2　分析和实验中使用的符号

名称	描述
C_{ser}	服务器端的计算代价
C_{auth}	客户端的验证代价
C_{tran}	传输代价
ΔC	代价差值
N_i^D	数据对象的数目（i 表示 fVO，uVO）
N_i^M	实体对象的数目（i 表示 fVO，uVO）
$fVOTime$	服务器计算 $fVOTime$ 的时间
$IZTime$	服务器计算 IZ 的时间
$IZvexNum\ (iz_n)$	服务器端产生 IZ 顶点数目
$result$	服务器端产生 $result$ 对象数目
$fVONum\ (f_n)$	服务器端产生 fVO 对象数目
$uVONum\ (u_n)$	服务器端产生 uVO 对象数目

IZ-Auth 和 UC-Auth 方法的总代价包括 C_{ser} 和 C_{tran} 以及 C_{auth} 三个方面. 我们从这三个方面来分析 IZ-Auth 和 UC-Auth 方法的性能及其适用性.

首先，从第三方服务器计算开销来考虑两种验证方法的性能. 若服务器采用 IZ-Auth 方法，则需要计算 IZ，k-depth contour 和 fVO 可离线生成，$result$ 和 uVO 可借助 IZ 并行计算出来. 若服务器采用 UC-Auth 方法，则需要计算 k-depth contour、fVO、$result$、uVO 信息. 因此，这两种验证方法在第三方服务器端的 ΔC_{ser} 为 C_{IZ}，其中服务器计算 IZ 的时间复杂度为 $O(k\cdot N_{fVO}^D\cdot N_{fVO}^D)$[21].

然后，从传输代价层面对比两种验证方法的性能. IZ-Auth 方法的通信量包括 f_n，u_n，iz_n，$result$；而 UC-Auth 方法的通信量包括 f_n，u_n，$result$，两者间近似 ΔC_{tran} 为 iz_n. 这说明 UC-Auth 方法在服务器计算和通信开销方面通常要胜于 IZ-Auth 方法. 实验表明，当数据量过万时，IZ-Auth 方法的传输量几乎等同于 UC-Auth 方法，而且这两种验证算法的传输量约占 BF-Auth 的 5%，这充分降低了通信代价，在一定程度上，提高了客户端验证结果的效率.

最后，从客户端验证 RkFN 查询结果的时耗上比较两种验证方法的性能. 客户端采用 IZ-Auth 和 UC-Auth 方法都需要根据 fVO 和 uVO 来重塑 MR-tree 根摘要，以此保证数据来源的可靠性，其时间复杂度为 $O(f_n\log f_n+u_n\log u_n)$. 而 IZ-Auth 方法接下来要检验 IZ 区域和 $result$. 当验证 IZ 时，假设 IZ 有 iz_n 顶点和 iz_n 边，并 fVO 中每个数据对象与 q 形成的中垂线都相交于 IZ 的每条边，那么 IZ 每条边上最多会有 N_{fVO}^D 个交点，则客户端验证 IZ 的最大开销为 $iz_n\cdot f_n+iz_n\cdot N_{fVO}^D\cdot 4\cdot N_{fVO}^D$，验证 $result$ 的花销为 $u_n\cdot iz_n$. 若客户端采用 UC-Auth 方法检验 RkFN 的 $result$，在成功检验了根摘要之后，fVO 和 uVO 利用外围圆性质来检验 $result$，该部分验证的最少代价是 uVO 中所有矩形的外围圆并集包含的对象数都不小于 k，即为 $N_{uVO}^D\cdot f_n+N_{uVO}^M\cdot N_{fVO}^D$；而最糟糕的验证代价是每个矩形外围圆并集包含的对象数都小于 k，假设每个矩形外围圆并集包含的对象数为 0，则最糟糕的验证开销为 $u_n\cdot f_n+N_{uVO}^M\cdot N_{fVO}^D\cdot N_{fVO}^D$. IZ-Auth 与 UC-Auth 方法的 ΔC_{auth} 为 $(iz_n\cdot f_n+iz_n\cdot N_{fVO}^D\cdot 4\cdot N_{fVO}^D+u_n\cdot iz_n)-(u_n\cdot f_n+N_{uVO}^M\cdot N_{fVO}^D\cdot N_{fVO}^D)=iz_n\cdot(f_n+u_n)-f_n\cdot u_n+(4\cdot iz_n-N_{uVO}^M)\cdot(N_{fVO}^D\cdot N_{fVO}^D)$. 当 k 取值较小时，fVO 和 uVO 中包含更多的 N_{fVO}^M 和 N_{uVO}^M，这缩减了检验 $result$ 的 VO，降低了客户端验证开销，结合上面的代价分析，IZ-Auth 方法的验证效率比

UC-Auth 方法要高，但随着集合 F 的数据量增加，第三方服务器采用 IZ-Auth 方法生成 IZ 的开销会随之增大，这说明服务器采用 IZ-Auth 方法的计算代价比 UC-Auth 高昂，但是客户端利用 IZ-Auth 验证 *result* 的代价却比 UC-Auth 低廉. 综上，IZ-Auth 和 UC-Auth 的总开销要远小于 BF-Auth，而且 IZ-Auth 和 UC-Auth 的传输量明显比 BF-Auth 少，这充分降低了用户流量开销. 由于 UC-Auth 有较低的服务器端开销，更适合 k 值较小或者小型数据库的验证，而 IZ-Auth 有较高的客户端验证效率，适合中小型数据库或者 k 值较小的大型数据库验证.

5 实验结果与分析

本文提出了两种有效的验证算法，一种是由 IZ 间接检验 RkFN 查询结果的 IZ-Auth 方法；另一种是通过 fVO 和 uVO 互检机制来验证结果的 UC-Auth 方法. 这两种算法采用了 C++ 语言实现，实验所使用的 PC 条件是 Intel(R) Core(TM) i7-3770 CPU@3.40 GHz 处理器、双核、4 GB 运行内存. 实验中使用的真实数据集由 Wang 等人[11]提供，包含 476587 个数据点，由 4 个数据集构成，分别为加利福尼亚路网节点(CA)，洛杉矶路网(SF)，北美路网(NA)以及加利福尼亚 POI 点，真实数据集中的 $|F|$ 和 $|U|$ 表示从真实数据集中抽取的数据点的个数，分别抽取 1×10^4，2×10^4，3×10^4，5×10^4，7×10^4 和 1×10^5 个数据点进行实验. 为保证实验质量，合成数据集由随机生成 3×10^6 个数据点构成，基本属于均匀分布，合成数据集中的 $|F|$ 和 $|U|$ 表示从 3×10^6 个合成数据点中随机抽取的数据. 实验中使用的相关参数及取值参看表 3，表中被加粗的数字为实验参数的默认值.

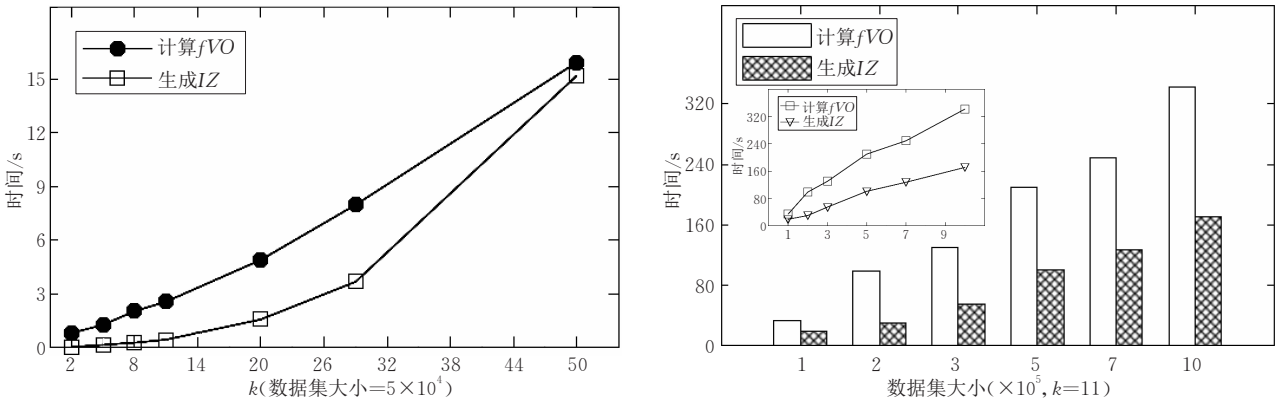
表 3 RkFN 查询验证的实验参数

参数	取值
k	2，5，8，**11**，20，29，50
real data set of F(×1000)	10，20，30，**50**，70，100
real data set of U(×1000)	10，20，30，**50**，70，100
Data Distribution	CA，SF，**NA**，POI of CA
synthetic data set of F(×1000)	100，200，300，**500**，700，1000
synthetic data set of U(×1000)	100，200，300，**500**，700，1000

5.1 第三方服务器计算代价

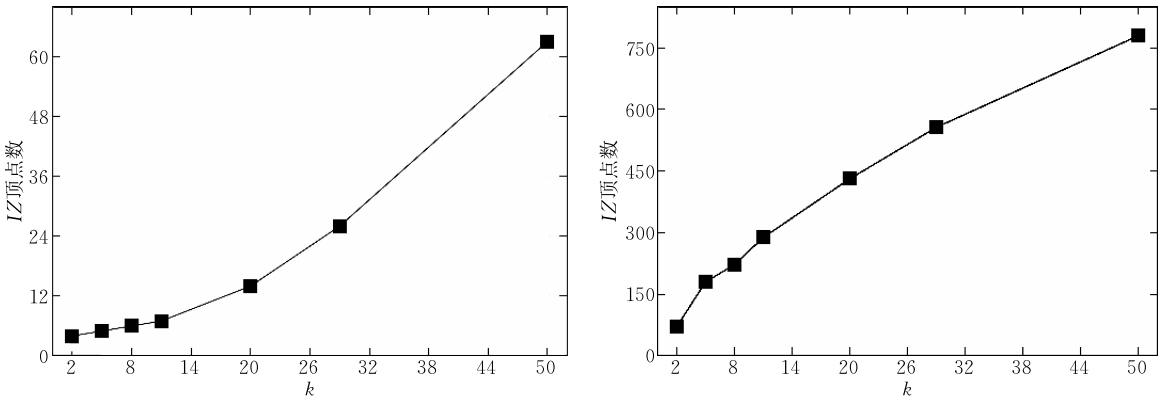
第三方服务器端的计算代价主要包括两个层面，一是计算 fVO 和 IZ 的时间花销，二是生成 VO 和结果的代价. 图 11 展示了计算 fVO 和 IZ 时间随 k 和数据集大小的增加而增加，原因参看 4.5 节的代价分析. 图 11(b)中的折线图表明了计算 fVO 和 IZ 的时间变化趋势，当数据量达到百万时，服务器生成 fVO 和 IZ 的时间将达到几分钟. 图 12(b)中合成数据集包含 5×10^5 个数据点，是图 12(a)中真实数据集的 10 倍，而且其 IZ 顶点数随 k 增加的速率约是真实数据集的 10 倍，这增加了服务器和客户端的计算压力，但并不能引起昂贵的通信代价，因为传输的 IZ 顶点数不到总 VO 的千分之一. 图 13 是 IZ 顶点数随着数据集大小变化，可以看出 IZ 顶点数的总量不到数据集的 1%，这侧面地反映出 IZ-Auth 方法的通信开销与 UC-Auth 方法基本相同，且 IZ 顶点数不会随着 k 和数据集无限增加，这是因为在给定数据集中，k-depth contour 是关于 k 值对称的. 假设空间中有 6 个点，那么 $k=2$ 和 $k=4$ 的 k-depth contour 相同.

第三方服务器端 I/O 访问代价和执行查询的 CPU 代价与 Wang 等人[11]论文中相近，这里不占用篇幅说明. 上面提到计算 fVO 和 IZ 时间不是服务器执行 RkFN 查询的总 CPU 花销，而是验证实验



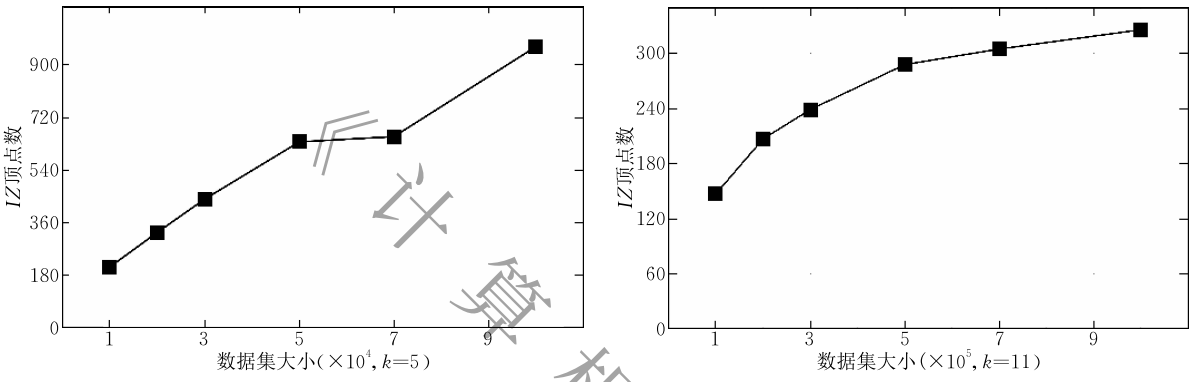
(a) 真实数据中随k变化　　(b) 合成数据中随数据集大小变化

图 11 生成 fVO 和 IZ 的时间随 k 和数据集大小改变



(a) 真实数据集　　(b) 合成数据集

图 12　IZ 顶点数随 k 变化



(a) 真实数据集　　(b) 合成数据集

图 13　IZ 顶点数随数据集大小变化

中额外生成的验证信息时间. 由于第三方服务器传输的是文本文件，故客户端的 I/O 访问代价可以忽略.

5.2　通信代价

通信代价将直接影响到用户流量的花销，实质是第三方服务器生成总 VO 以及结果所包含的对象数占用的流量开销. 本文提出的验证算法致力于缩减 VO 来降低通信代价. 图 14 中 BF 表示验证方法 BF-Auth 在各规模数据集下所产生的 VO. 由图可知，BF-Auth 产生的总 VO 比 IZ-Auth 和 UC-Auth 产生的多几倍，其中 IZ-Auth 算法产生的 IZ 顶点数没有明显地展现出来，参看图 12 和图 13，IZ-Auth 产生的 IZ 顶点数与 IZ-Auth 产生的 VO 相比，占极少比例，可以忽略不计，且 IZ-Auth 和 UC-Auth 产生的总 VO 呈平缓增长趋势，而 BF-Auth 呈指数上涨的趋势. 这主要是因为 fVO 包含对象个数与 k-depth contour 有直接关系，当 k 取值较小时，k-depth contour 外部的对象数目相对整个数据集而言可忽略不计. uVO 包含对象个数是由结果对象所在矩形多少决定的. 而 BF-Auth 的总 VO 是直接随着数据集的变化而变化.

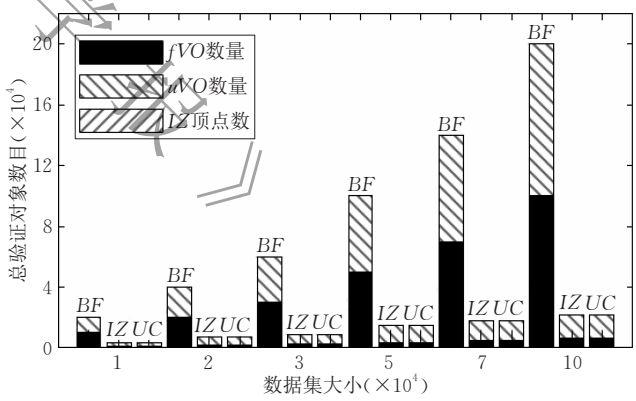
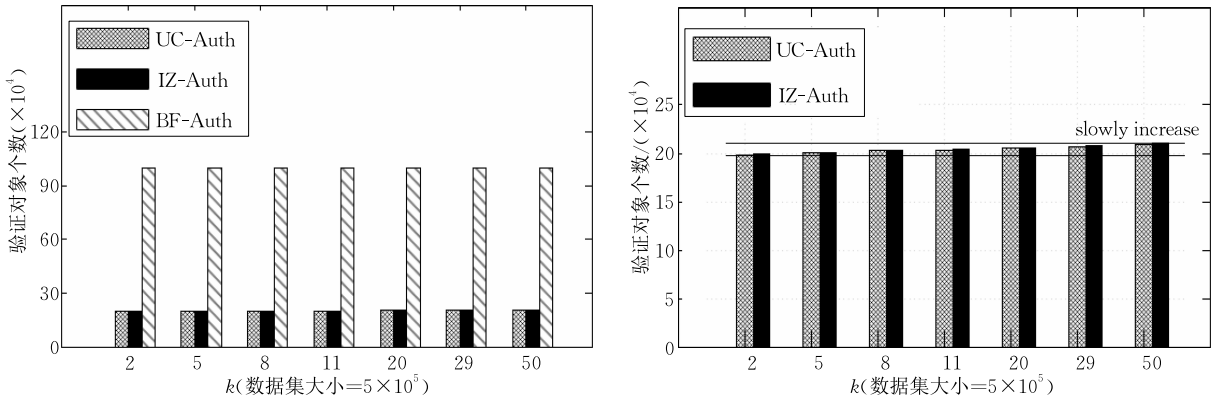


图 14　真实数据中三种验证算法总 VO 随数据集大小变化

图 15 是合成数据集为 5×10^5 时，三种验证算法产生的总 VO 随 k 的变化. 图 15(a)中 IZ-Auth 方法产生的总 VO 不受 IZ 顶点数影响，与 UC-Auth 生成的验证对象大小基本相等，约占 BF-Auth 方法的总 VO 的五分之一，这说明了缩减 VO 能降低传输代价，减少用户流量损失；图 15(b)是对图 15(a)中的 IZ-Auth 和 UC-Auth 两种验证方法产生的验证对象间的差值(即 IZ 顶点数)进行了放大显示. 两种验证算法之间产生的 IZ 顶点数随 k 的增加而微微增多，这说明了 IZ 顶点数不足以提升总通信



(a) 三种验证方法产生的总VO随k变化 (b) IZ-Auth和UC-Auth产生的总VO随k变化的差异

图 15 合成数据集中三种验证方法的总 VO 随 k 变化

代价,因此,可以得出一个结论:IZ-Auth 的传输代价和 UC-Auth 的传输代价可以等同.

图 16 展示了真实数据集为 50 000 时的通信量变化.其中 fVO 和 uVO 以及结果对象数量随 k 的增加而增多.图 17 是结果和 uVO 对象个数与集合 U 间的对比趋势.在 k 值给定的情况下,图 17(a)和图 17(b)分别揭示了在真实数据集和合成数据集中结果和 uVO 对象个数与 U 之间有明显的数值差距,结果和 uVO 对象个数占 U 总数的 5%左右,IZ-Auth 和 UC-Auth 算法能压缩 VO,降低通信代价.

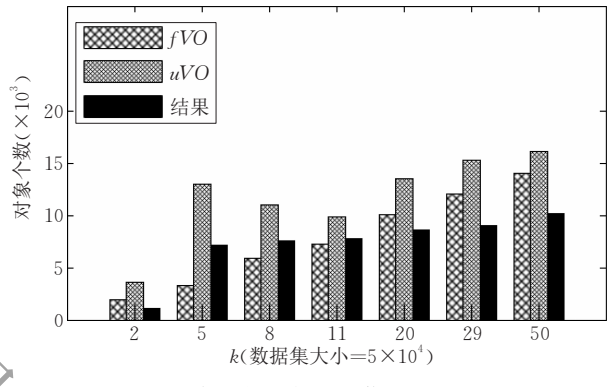
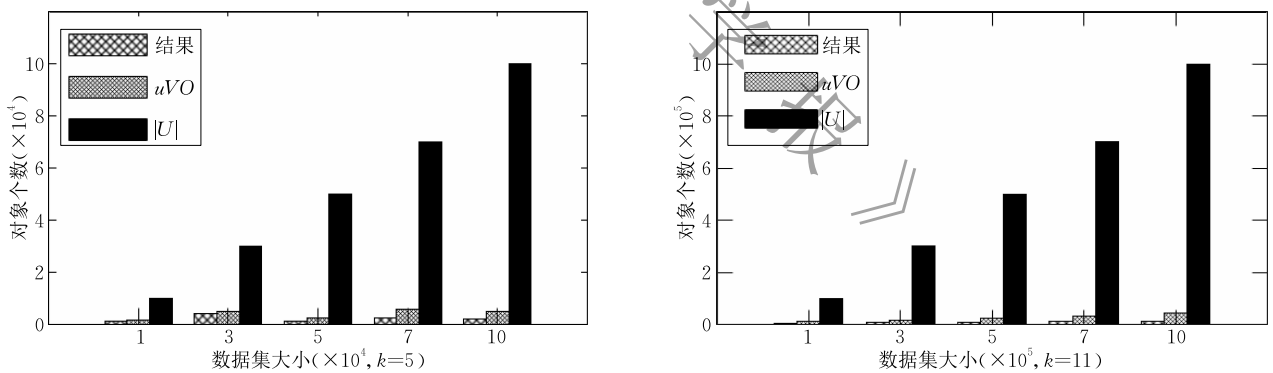
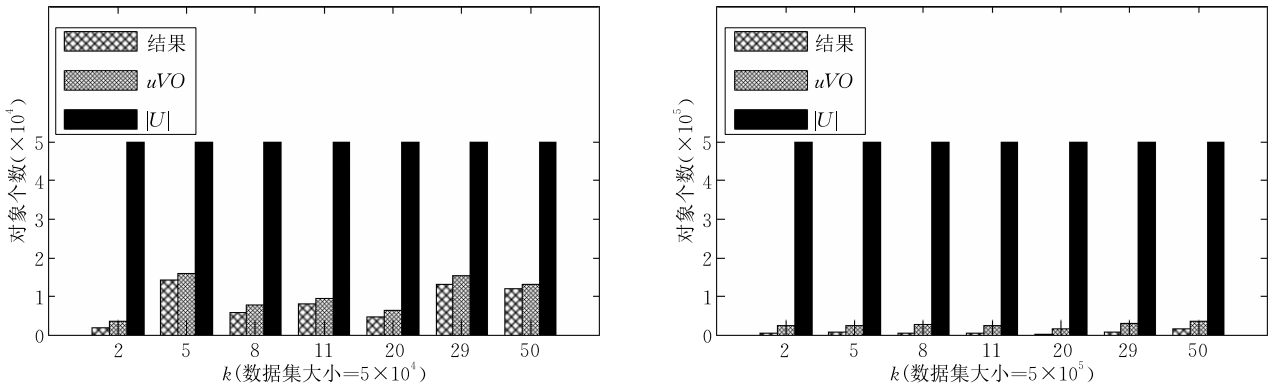


图 16 真实数据集中通信量随 k 变化



(a) 各个参数的对象数量随着真实数据集大小变化 (b) 各个参数的对象数量随着合成数据集大小变化



(c) 真实数据中各个参数的对象数量随着k变化 (d) 合成数据中各个参数的对象数量随着k变化

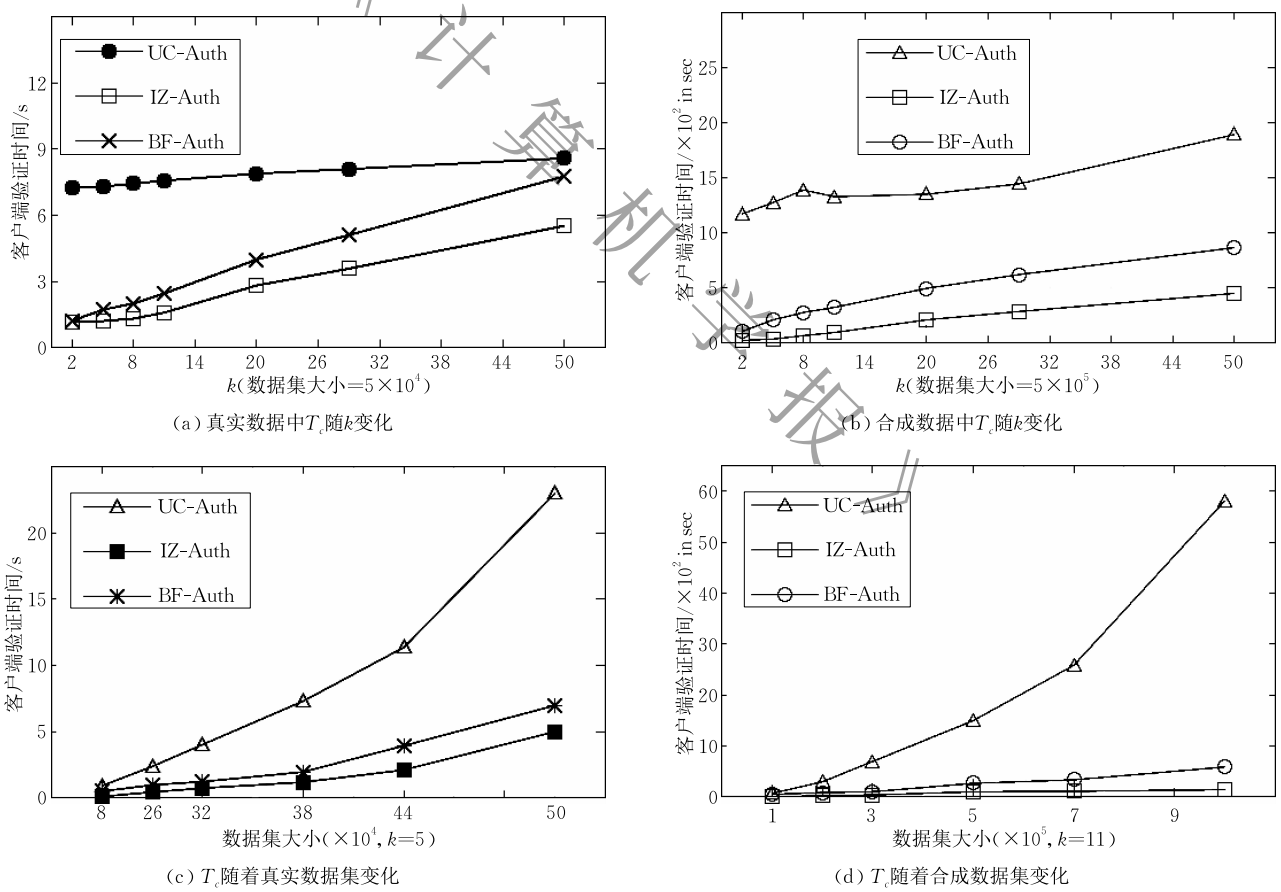
图 17 结果和 uVO 对象数量与 U 随 k 和数据集大小变化

在 k 值给定的情况下，结果和 uVO 对象个数并没有随着数据集的增加而显著增加，这说明了在 k 值较小时，数据集越大，缩减 VO 就越明显，客户端验证结果的效率就越高效. 图 17(c)和图 17(d)是在给定的真实和合成的数据集中，结果和 uVO 对象个数随着 k 的增加而缓慢增多. 相比 $|U|$，uVO 对象个数能明显地降低验证对象的传输量，但是缩减集合 U 的比例要视具体的数据分布和大小而定.

考虑到用户智能机的存储能力可能成为接受 VO 等数据的瓶颈，若第三方服务器采用 BF-Auth 方法来产生 VO，该 VO 的数量可能导致用户智能机面临着存储灾难，甚至之后需要超负荷验证. 为了避免上述问题发生，故本文提出了 IZ-Auth 和 UC-Auth 两种 RkFN 查询验证方法.

5.3　客户端验证代价

客户端验证代价是用户直接关注的问题. 如图 18 所示，三种验证算法的验证时间 T_c 随着数据集和 k 的变化趋势，其中 IZ-Auth 和 UC-Auth 方法的验证效率优于 BF-Auth 验证算法，且 IZ-Auth 比 UC-Auth 具有更明显的优势，原因是验证 fVO 中 MBRs 时，如果外围圆并集包含的对象数达到 k，该矩形就被成功验证，但当外围圆并集包含的对象数不到 k 时，该矩形实体的验证代价会增加，需要通过包围圆并集内的所有设施对象来进一步检验，这导致了 fVO 中部分对象被重复使用，但 UC-Auth 算法仍比暴力的 BF-Auth 有优势，该优势是有效降低通信代价. 图 18 展示了 BF-Auth 方法在真实数据中的验证时间约达 20 min，在合成数据中的验证时间达到小时级别，而 IZ-Auth 和 UC-Auth 方法的验证效率要明显高于 BF-Auth 方法，这能提升良好的用户体验.



(a) 真实数据中 T_c 随 k 变化　(b) 合成数据中 T_c 随 k 变化

(c) T_c 随着真实数据集变化　(d) T_c 随着合成数据集变化

图 18　验证方法的验证时间随 k 和数据集大小的变化

真实数据和合成数据所表现出的通信代价和客户端验证效率有较大差异性，主要是因为合成数据集的数据量较大；此外，真实数据的分布呈现一定的偏斜，也有助于剪枝更多的数据.

综上，相比 BF-Auth，UC-Auth 和 IZ-Auth 展现了其更高的验证效率. 通常，IZ-Auth 方法的验证效率高于 UC-Auth 方法，其原因是 IZ-Auth 利用 fVO 检验 IZ 区域完整性时，由于 IZ 顶点数较少，故区域验证代价并不昂贵，接着用被验证的 IZ 筛选 uVO，该筛选过程比计算至少 k 次的过程更节省

时间，所以，通常情况下，客户端采用 IZ-Auth 方法的验证代价要低于 UC-Auth 方法. 在实际生活中，基于位置服务的查询需求越来越被人类重视，用户更加关注时间效率. 当外包数据库中数据量不超过 3×10^5 时，第三方服务器生成 IZ 顶点数的时间代价并不高昂，当数据量超过百万级别时，服务器生成 IZ 顶点数将相当耗费时间，所以，针对中小型数据集的验证，IZ-Auth 和 UC-Auth 方法都比较适用.

6 总 结

本文首次针对外包数据库中的反向 k 最远邻问题进行了研究，提出了 IZ-Auth 和 UC-Auth 验证方法. IZ-Auth 方法是利用 IZ 来间接地验证 RkFN 查询结果，最主要的验证环节是检验 IZ 是否完整，从而保证结果的正确性. UC-Auth 方法是通过验证对象间的互检机制来判断结果的正确性，该互检机制采用了外围圆的特性来验证结果的三性是否正确. 这两种验证方法的优势是缩减了验证对象的大小，降低了通信代价以及客户端验证开销等. 通过这两种验证算法，用户不仅可以获得满足要求的结果，而且可以保证获取结果的正确性. 本文通过大量的真实数据和合成数据来检验了这两种验证方法，实验证明了 RkFN 查询验证算法的有效性和实用性.

参 考 文 献

[1] Yang Yin, Papadopoulos S, Papadias D, et al. Authenticated indexing for outsourced spatial databases. The Very Large Data Base Journal, 2009, 18(3): 631-648

[2] Chen Qian, Hu Hai-Bo, Xu Jian-Liang. Authenticating top-k queries in location-based services with confidentiality. Proceedings of the Very Large Data Base Endowment, 2014, 7(1): 49-60

[3] Lin Xin, Xu Jian-Liang, Hu Hai-Bo. Authentication of location-based skyline queries//Proceedings of the 20th ACM Conference on Information and Knowledge Management. Glasgow, UK, 2011: 1583-1588

[4] Lin Xin, Xu Jian-Liang, Hu Hai-Bo, et al. Authenticating location-based skyline queries in arbitrary subspaces. IEEE Transactions on Knowledge and Data Engineering, 2014, 26(6): 1479-149

[5] Wu Ding-Ming, Choi B, Xu Jian-Liang, et al. Authentication of moving top-k spatial keyword queries. IEEE Transactions on Knowledge and Data Engineering, 2015, 27(4): 922-935

[6] Yiu M, Lo E, Yung D. Authentication of moving knn queries//Proceedings of the 27th International Conference on Data Engineering. Hannover, Germany, 2011: 565-576

[7] Jing Yi-Nan, Hu Ling, Ku W, et al. Authentication of k nearest neighbor query on road networks. IEEE Transactions on Knowledge and Data Engineering, 2014, 6(6): 1494-1506

[8] Yiu M, Lin Yi-Min, Mouratidis K. Efficient verification of shortest path search via authenticated hints//Proceedings of the 26th International Conference on Data Engineering. California, USA, 2010: 237-248

[9] Li Guo-Hui, Luo Chang-Yin, Li Jian-Jun. Authentication of reverse k nearest neighbor query//Lecture Notes in Computer Science 9049, Berlin: Springer, 2015: 625-640

[10] Chen Zi-Jun, Hong Ji-Hai, Liu Wen-Yuan. Authentication of reverse k nearest neighbor query verification in outsourcing spatial database. Journal of Chinese Computer Systems, 2013, 34(8): 1819-1824(in Chinese with English Abstract)
(陈子军，洪济海，刘文远. 外包空间数据库中反向 k 近邻查询验证. 小型微型计算机系统，2013，34(8)：1819-1824)

[11] Wang Shen-Lu, Cheema M, Lin Xue-Min, et al. Efficiently computing reverse k furthest neighbors//Proceedings of the 32nd International Conference on Data Engineering. Helsinki, Finland, 2016: 1110-1121

[12] Yao Bin, Li Fei-Fei, Kumar P. Reverse furthest neighbors in spatial databases//Proceedings of the 25th International Conference on Data Engineering. Shanghai, China, 2009: 664-675

[13] Liu Jian-Quan, Chen Han-Xiong, Furuse K, et al. An efficient algorithm for reverse furthest neighbors query with metric index//Lecture Notes in Computer Science 6262. Berlin, Germany: Springer, 2010: 437-451

[14] Xu Xiao-Jun, Bao Jin-Song, Yao Bin, et al. Reverse furthest neighbors query in road networks. Journal of Computer Science and Technology, 2017, 32(1): 155-167

[15] Li Bo-Han, Zhang Chao, Chen Wei-Tong, et al. Dynamic reverse furthest neighbor querying algorithm of moving objects//Proceedings of the 12th International Conference on Advanced Data Mining and Applications. QLD, Australia, 2016: 266-279

[16] Yang Yin, Papadopoulos S, Papadias D, et al. Spatial outsourcing for location-based services//Proceedings of the 24th International Conference on Data Engineering. Cancún, México, 2008: 1082-1091

[17] Narasimha M, Tsudik G. Authentication of outsourced databases using signature aggregation and chaining//Lecture Notes in Computer Science 3882. Berlin, Germany: Springer, 2006: 420-436

[18] Cheema M, Shen Zhi-Tao, Lin Xue-Min, et al. A unified framework for efficiently processing ranking related queries//Proceedings of the 17th International Conference on Extending Database Technology. Athens, Greece, 2014: 427-438

[19] Cheema M, Lin Xue-Min, Zhang Wen-Jie, et al. Influence zone: Efficiently processing reverse k nearest neighbors queries//Proceedings of the 27th International Conference on Data Engineering, Hannover, Germany, 2011: 577-588

[20] Wu Wei, Yang Fei, Chan C, et al. FINCH: Evaluating reverse k-nearest-neighbor queries on location data. Proceedings of the Very Large Data Base Endowment, 2008, 1(1): 1056-1067

[21] Yang Shi-Yu, Cheema M, Lin Xue-Min, et al. Reverse k nearest neighbors queries and spatial reverse top-k queries. The Very Large Data Base Journal, 2017, 26(2): 151-176

WANG Hai-Xia, born in 1991, M. S. candidate. Her research interest is spatial data management.

GU Yu, born in 1981, Ph. D., professor, Ph. D. supervisor. His current research interests include spatial data management, graph data management and big data analysis.

YU Ge, born in 1962, Ph. D., professor. His research interests include distributed and parallel database, OLAP and data warehousing, data integration, graph data management, etc.

Background

As data increasingly grows, Data Owners, which are constrained by storage capacity or computing power, cannot efficiently handle user requests. Therefore, delegating data to Third-Party Data Publishers, which manage the data and execute user queries, has become an inevitable trend of outsourcing spatial databases. However, the results returned by the Third-Party may be tampered or lost, so the Client needs to verify the correctness of the results, in order to decide whether to accept the query results from the Third-Party.

In recent years, the query authentication of outsourcing spatial databases has aroused widespread concerns of researchers. Specifically, the authentication of the range query, k nearest neighbor query, spatial-keyword queries and reverse k nearest neighbor query has been explored. However, the authentication of the reverse k furthest neighbor (RkFN) query has not been sufficiently studied. The RkFN neighbor queries have very important applications in complex spatial analysis and location selection, and effective authentication technique is demanded for the outsourcing scenario.

To crack this nut for the first time, this paper designs two effective authentication algorithms for RkFN, namely IZ-Auth and UC-Auth. By optimizing the large-scale verification objects, they can efficiently reduce the cost of both communication and client-side verification. The client-side authentication of IZ-Auth involves the following techniques: reconstructing the root to judge whether VO is reliable, IZ theorems to check the integrity of IZ, and authentication of range query. IZ-Auth can gain a high validation efficiency, which can fully reduce the cost of communication and client verification. Furthermore, UC-Auth is proposed based on the characteristics of the outer circle, which does not need to calculate the IZ boundary and thus reduces the overhead of the server computing and the communication cost of transporting, but at the cost of increasing the client verification cost. So the two authentication methods can be supplementary to each other in the real applications according to different user requirements. The effectiveness and efficiency of the proposed methods have been verified by the extensive experiments.

This paper is supported by the National Natural Science Foundation of China (No. 61472071, No. 61433008), the Fundamental Research Funds for the Central Universities (No. N171605001), and the Natural Science Foundation of Liaoning Province(No. 2015020018).

The authors of this paper have been engaged in spatial data management. They have published some papers in this area.