

一种实时监控最近邻的近似算法*

金澈清¹⁺, 崇志宏², 周傲英³

JIN Cheqing¹⁺, CHONG Zhihong², ZHOU Aoying³

1.华东理工大学 计算机科学与工程系, 上海 200237

2.东南大学 计算机科学与工程学院, 南京 211189

3.复旦大学 计算机科学与工程系, 上海 200433

1.Department of Computer Science, East China University of Science and Technology, Shanghai 200237, China

2.School of Computer Science and Engineering, Southeast University, Nanjing 211189, China

3.Department of Computer Science and Engineering, Fudan University, Shanghai 200433, China

+Corresponding author: E-mail: cqjin@ecust.edu.cn

JIN Cheqing, CHONG Zhihong, ZHOU Aoying. An approximate approach to monitoring nearest neighbors in real time. Journal of Frontiers of Computer Science and Technology, 2007,1(2):146-159.

Abstract: The biggest challenge to processing high-speed data over distributed environment is to output qualified results by using small amount of network resource. The paper studies how to cope with nearest neighbors query over distributed environment and proposes a novel solution, which is capable of answering not only precise query, but also five kinds of approximate queries. After installing a Smart Filter in each remote site to filter parts of incoming data, the novel approach continuously adjusts the range monitored by each filter to reduce the overall communication cost. Theoretic analysis and experimental results based on synthetic datasets and real dataset indicate that new approach owns good performance.

Key words: nearest neighbors query; distributed environment; approximate algorithm; data stream

摘要: 处理分布式环境下高速数据的最大挑战在于如何利用少量网络资源输出高质量的查询结果。对面向分布式环境的最近邻查询问题进行了研究,提出了一种基于过滤器的新方法,不仅能计算精确查询结果,还能够处理五类近似查询。该方法在各个远程站点均安装了智能过滤器,并通过合理设置过滤器的范围来降低数据传输量。理论分析及基于模拟数据集合和真实数据集合的实验报告均表明新方法具有较高的性能。

关键词: 最近邻查询; 分布式环境; 近似算法; 数据流

文献标识码:A **中图分类号:** TP301

* the Key Project of National Natural Science Foundation of China under Grant No.6049325,6049327(国家自然科学基金重大项目).

Received 2007-04, Accepted 2007-06.

1 引言

近年来,很多新型应用都涉及到如何实时处理高速数据,并获得查询结果,包括因特网应用、传感器网络、日志分析等。在这类应用中,数据产生速度极快,数据量极大。例如,因特网主干网的高速路由器的传输速度可高达数十字节每秒。为监控网络状态,各路由器持续产生日志记录,发送到一个中心主机;中心主机在线分析日志记录,计算特定查询目标值,评估网络状况。目前,针对这类问题的典型方案是设计资源节约型的单遍扫描算法,提供高质量的近似查询结果。重要的资源包括内存资源、计算资源和网络资源等^[1-3]。

最近邻查询(Nearest Neighbors query,简称 NN 查询)^[4-5]的目标是获取距某查询点最近的若干个对象。例如,一个出租车公司经营很多车辆,当车辆行驶时,其地理坐标不断变化。查询 Q_1 :“哪 5 辆车距离某城市地标最近?”就是一个典型的最近邻查询。一个现代化的化学工业区里布置着众多气体浓度检测装置,实时监测大气中危险气体的浓度。查询 Q_2 :“哪 3 个点的 SO_2 气体浓度最高?”是一个 top- k 查询^[6],同时也是一个最近邻查询,即:“返回距浓度值 ∞ 最近的 3 个点”。此外,大多数应用也容许近似查询结果,如查询 Q_3 :“请列举 5 辆车,它们都是距某地标最近的 8 辆车之一”。

一个简单的处理方法(Naïve approach)就是:每个远程站点(出租车或气体浓度检测装置)均频繁地向中心主机汇报所监测对象的当前值(各出租车的当前坐标值或气体浓度检测装置采集的气体浓度值),由中心主机进行计算。但这种方法的扩展性较弱,当对象数目增多时,所消耗的网络资源和计算资源急剧增加,从而降低了服务质量。基于门槛的方法(threshold-based approach)能显著降低资源耗费^[7-9]。这类方法在各远程站点安装一个过滤器,监控一个范围。当远程站点与查询点的距离在范围之内时,无需传送站点的位置;反之,远程站点马上向中心主机发送当前位置。例如,在一维空间中,假设查询点

(中心主机)的位置是 0,远程站点的过滤器范围是 [6,10],其运动轨迹是 {7,9,11},则仅 11 被传输到中心主机。中心主机可以根据各远程站点所监控的范围来计算查询结果。

但现有研究成果仍有一定的局限性。首先,现有工作未能支持多种查询类型。典型的近似最近邻查询包括绝对数值容许度最近邻查询、相对数值容许度最近邻查询、纳伪容许度最近邻查询、拒真容许度最近邻查询、排名容许度最近邻查询等(第 2.2 节)。但是,Mouratidis 方法^[8]仅支持精确查询,RTP 方法^[7]仅面向排名容许度最近邻查询,FT-NRP 方法^[7]仅面向拒真容许度和纳伪容许度近似查询,Jin 方法^[9]仅面向精确查询、绝对数值容许度最近邻查询和排名容许度最近邻查询。其次,部分工作不支持移动查询点。在通用场景中,远程对象和中心主机均不断移动,然而 RTP 方法和 FT-NRP 方法都不支持这种场景。再次,部分工作未能统一考虑多查询处理情况,仅循环独立处理单个查询,例如 RTP 方法、FT-NRP 方法和 Jin 方法等。

文章提出一个新方法,在各个远程站点布置一个新型过滤器(SF),它仅向中心主机发送部分数据;中心主机根据各过滤器的参数设置来计算近似查询结果。相比现有研究成果,新方法具备以下四点贡献。

- (1)支持多种查询类型,包括精确查询和五类近似查询。
- (2)能够处理中心主机不断移动(或静止)的情况。
- (3)新方法统一处理在中心主机注册的多个查询,提高网络资源利用率。
- (4)理论分析和实验结果均表明,新方法的网络传输开销更小。

第 2 章定义了系统模型和六种最近邻查询;第 3 章提出一种新型数据结构,即智能过滤器,并基于该结构设计一种针对静止查询点的新方法;第 4 章和第 5 章分别扩展基本方法,讨论如何处理移动查询点情况和多最近邻查询情况;第 6 章通过一系列实

验报告,充分验证了新方法的性能;第7章介绍部分相关工作;第8章简要地总结了工作内容。

2 模型和查询定义

2.1 模型

系统模型如图1所示,包含 n 个远程站点(remote sites)和1个协调者站点(coordinator site,也就是中心主机,或查询点)。各远程站点 i 监控一个对象 O_i 。令 $V_{i,t}$ 表示对象 O_i 在 t 时刻的位置,易知,对象 O_i 的运动轨迹为: $\{V_{i,0}, V_{i,1}, \dots\}$ 。协调者站点 q 可以移动,其运动轨迹可以表示为: $\{q_1, q_2, \dots\}$ 。令 $S = \{O_1, O_2, \dots, O_n\}$, 包含所有对象。各远程站点之间无法相互通讯,只能和协调者站点通讯。用户可以在协调者站点注册(多个)最近邻查询,实时监控结果。

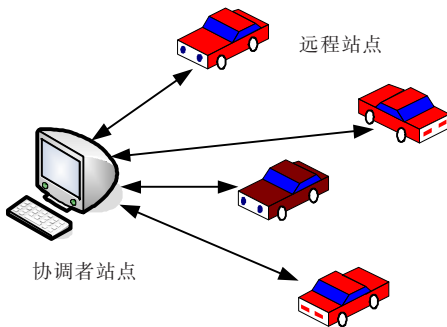


图1 系统模型图

Fig.1 The model of system

各对象的位置可以 d -维空间表示。函数 $dist$ 表示两个位置之间的距离,性质如下(其中, V_1, V_2 和 V_3 是3个对象的位置)。

性质1 $dist(V_1, V_2) = dist(V_2, V_1)$

性质2 $dist(V_1, V_2) > 0 (V_1 \neq V_2)$, 且 $dist(V_1, V_2) = 0 (V_1 = V_2)$

性质3 $dist(V_1, V_2) \leq dist(V_2, V_3) + dist(V_1, V_3)$

2.2 最近邻查询

文章研究精确最近邻查询和五类近似最近邻查询,分别如定义1-定义6所示。

定义1 精确最近邻查询($kNN(q, k)$)。令 q 表示查询点, k 表示对象数目。在任一时刻 t , 返回集合 A_t , 满足下列条件:(1) $|A_t| = k$; (2) $\forall O_i \in A_t, O_j \in S - A_t$,

使得 $dist(V_{i,t}, q_t) \leq dist(V_{j,t}, q_t)$ 。

定义2 绝对数值容许度最近邻查询($a-kNN(q, k, e)$)。令 q 表示查询点, k 表示对象数目, e 表示误差参数。在任一时刻 t , 返回集合 A_t , 满足下列条件:(1) $|A_t| = k$; (2) $\forall O_i \in A_t$, 使得 $dist(V_{i,t}, q_t) \leq e + \max_{O_j \in kNN(q_t, k)} (dist(V_{j,t}, q_t))$ 。

定义3 相对数值容许度最近邻查询($v-kNN(q, k, e)$)。令 q 表示查询点, k 表示对象数目, e 表示误差参数。在任一时刻 t , 返回集合 A_t , 满足下列条件:(1) $|A_t| = k$; (2) $\forall O_i \in A_t$, 使得 $dist(V_{i,t}, q_t) \leq (1+e) \max_{O_j \in kNN(q_t, k)} (dist(V_{j,t}, q_t))$ 。

定义4 纳伪容许度最近邻查询($p-kNN(q, k, e^+)$)。令 q 表示查询点, k 表示对象数目, e^+ 表示最大纳伪比率参数。在任一时刻 t , 返回集合 A_t , 满足下列条件:(1) $|A_t| = k$; (2) $\frac{|A_t \cap kNN(q_t, k)|}{k} \geq 1 - e^+$ 。

定义5 拒真容许度最近邻查询($n-kNN(q, k, e^-)$)。令 q 表示查询点, k 表示对象数目, e^- 表示最大拒真比率参数。在任一时刻 t , 返回集合 A_t , 满足下列条件:(1) $|A_t| = k$; (2) $\frac{|A_t \cap kNN(q_t, k)|}{k} \geq 1 - e^-$ 。

由定义4和5可知, 纳伪容许度最近邻查询和拒真容许度最近邻查询可以相互转化。

定义6 排名容许度最近邻查询($r-kNN(q, k, e)$)。令 q 表示查询点, k 表示对象数目, e 表示误差参数。在任一时刻 t , 返回集合 A_t , 满足下列条件:(1) $|A_t| = k$; (2) $A_t \subseteq kNN(q_t, k+e)$ 。

例1 某应用含12个远程对象和1个协调者站点,且它们在某一时刻的位置如图2所示。易知: $kNN(q, 5) = \{O_1, O_2, O_3, O_4, O_5\}$; $a-kNN(q, 5, 1)$ 的答案可以是 $\{O_1, O_2, O_3, O_4, O_5, O_6\}$ 中的任意5个对象组成的子集; $v-kNN(q, 5, 0.5)$ 的答案可以是 $\{O_1, O_2, O_3, O_4, O_5, O_6, O_{11}\}$ 中的任意5个对象组成的子集; $p-kNN(q, 5, 0.2)$ 或 $n-kNN(q, 5, 0.2)$ 的答案可以是 $\{O_1, O_2, O_3, O_4, O_5\}$ 中的任意4个对象加上任一其他对象; $r-kNN(q, 5, 2)$ 的答案可以是 $\{O_1, O_2, O_3, O_4, O_5, O_6, O_{11}\}$ 中的任意5个对象组成的子集。

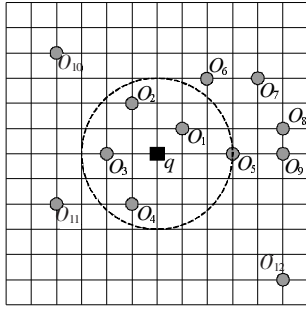


图 2 例 1 中各对象的位置分布

Fig.2 Positions of objects in Example 1

3 面向固定协调者站点的单查询处理方法

该章研究一个简单应用场景,协调者站点静止不变,且仅处理单个最近邻查询。

各个远程站点 i 均安装一个智能过滤器 SF_i (第 3.1 节), 监控一个范围。若当前位置 $V_{i,t}$ 距协调者站点 q_t 的值在监控范围之内, 不发送数据; 反之, 站点 i 调用函数 $handleV$ 调整监控范围, 并报告当前位置 $V_{i,t}$ 。协调者站点备份了所有过滤器设置。当远程站点调用 $handleV$ 调整过滤器设置, 并发送 $V_{i,t}$ 给协调者站点时, 协调者站点也调用 $handleV$ 函数, 同步更新过滤器设置。在任一时刻, 若协调者站点无法根据当前过滤器设置计算查询结果, 则调用 $rearrange$ 函数重新设置各过滤器(包括本地保存的和远程站点的)。协调者站点可根据过滤器设置参数计算查询结果。

第 3.1 节描述一个新的过滤器, 称为智能过滤器(Smart Filter); 第 3.2 节阐述协调者站点如何利用过滤器参数计算查询结果; 第 3.3 节描述远程站点和协调者站点的处理过程; 最后在 3.4 节分析新方法的性能。

3.1 智能过滤器(Smart Filter)

一个智能过滤器 $SF(c, l, u, \psi, L, U)$ 含 6 个参数。字段 c 是协调者站点的位置, 当协调者站点固定时, c 值不变。字段 l 和 u 描述它的监控范围。令 V 表示远程站点的当前位置, 当距离 $dist(V, c) \in [l, u]$ 时, 无需向协调者站点报告新位置。松弛系数 ψ 用于辅助更新 l 和 u 值, 它由协调者站点指定(第 3.4.3 节)。参数 L 和 U 由协调者站点指定, 辅助更新 l 和 u 值。当距离 $dist(V, c) \notin [l, u]$ 时, 智能过滤器 SF 调用函数 $handleV$ (算法 1) 调整过滤器参数。

算法 1 $handleV(V, c, \psi, L, U)$

- (1) IF ($dist(V, c) > U$) THEN
- (2) $L=U; U=\infty; l=L; u=U;$
- (3) ELSEIF ($dist(V, c) < L$) THEN
- (4) $U=L; L=0; l=L; u=U;$
- (5) ELSE
- (6) IF ($L \leq dist(V, c) \leq (1+\psi)L$) THEN $l=L;$
 ELSE $l=(L+dist(V, c))/2;$
- (7) IF ($(1-\psi)U \leq dist(V, c) \leq U$) THEN $u=U;$
 ELSE $u=(U+dist(V, c))/2;$

由算法 1 可知, $[L, U] \supseteq [l, u]$ 。当 $u=\infty$ 或者 $l=0$ 时, 监控范围分别是 $[l, \infty]$ 或 $[0, u]$, 过滤器表现为单边受限。图 3 描述了算法 1 调整过滤器参数时的三种典型情况。无阴影的圆圈(D_{t-1})和有阴影的圆圈(D_t)分别代表在时间 $t-1$ 和 t 时对象距协调者站点的距离。当 D_t 超出 U 时(图 3(a)), 监控范围被重新设置为 $[U, \infty]$; 当 D_t 位于 $[(1-\psi)U, U]$ 时(图 3(b)), 监控范围被重新设置为 $[l, U]$; 当 D_t 位于 $[(1+\psi)L, l]$ 时(图 3

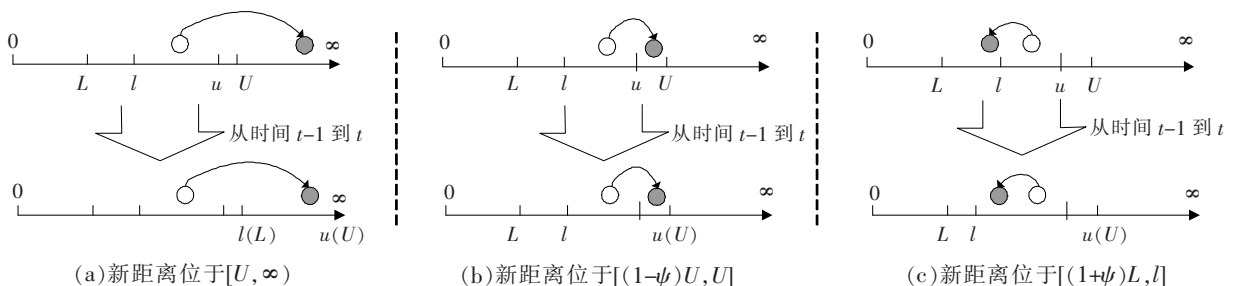


图 3 $handleV$ 调整过滤器范围 $[l, u]$ 的典型情况

Fig.3 Typical situations for subroutine $handleV$ to adjust the range $[l, u]$

(c)), 监控范围被重新设置为 $[\frac{L+D_i}{2}, u]$ 。

3.2 基于智能过滤器参数设置计算查询结果

协调者站点保存所有过滤器的参数设置, 保证各对象(如 O_i)至协调者站点的距离位于 $[l_i, u_i]$ 之间。在一定条件下, 协调者站点能够基于这些过滤器参数设置来计算查询结果。

令 d_i^- 和 d_i^+ 分别表示对象 O_i 至协调者站点的最小和最大距离。当协调者站点固定时, 有如下公式:

$$d_i^- = l_i \quad (1)$$

$$d_i^+ = u_i \quad (2)$$

令集合 Y_i 包含所有比 O_i 更接近协调者站点的对象, R_i 表示集合 Y_i 的大小的最大可能值, $R_i = |Y_i|$ 。若仅依据过滤器参数设置, 则 Y_i 包含所有满足 $d_j^- \leq d_i^+$ 的对象 O_j 。即:

$$R_i = |Y_i|$$

其中 $Y_i = \{O_j | (O_j \in S) \wedge (d_j^- < d_i^+)\}$ (3)

定理 1 若集合 A 满足: (1) $A = \{O_i | R_i < k\}$; (2) $|A| = k$, 则 A 是精确最近邻查询 $kNN(q, k)$ 的答案。

证明 由于 $R_i < k$, 说明比 O_i 更近的对象不足 k 个, 即 O_i 是一个 k -最近邻。又由于 $|A| = k$, 因此集合 A 是查询答案。得证。

令 D_k 表示第 k 个最小的 d_i^- 值。

定理 2 若集合 A 满足 (1) $|A| = k$; (2) $\forall O_i \in A$, 有 $d_i^+ < D_k + e$, 则 A 是绝对数值容许度最近邻查询 $a-kNN(q, k, e)$ 的答案。

定理 3 若集合 A 满足 (1) $|A| = k$; (2) $\forall O_i \in A$, 有 $d_i^+ < (1+e)D_k$, 则 A 是相对数值容许度最近邻查询 $v-kNN(q, k, e)$ 的答案。

由于 D_k 是协调者站点 q 的 k -最近邻的最小可能距离, 根据定义 2 和定义 3, 定理 2 和定理 3 得证。

定理 4 若集合 A 满足 (1) $|A| = k$; (2) $\frac{|A \cap kNN(q, k)|}{k} \geq 1 - e^+$, 则 A 是纳伪容许度最近邻查询 $p-kNN(q, k, e^+)$ 的答案。

定理 5 若集合 A 满足 (1) $|A| = k$; (2) $\forall O_i \in A$, 有 $O_i \in kNN(q, k+e)$, 则 A 是排名容许度最近邻查询 $r-kNN(q, k, e)$ 的答案。

定理 4 和定理 5 可分别根据定义 4~定义 6 直接得证。

3.3 远程站点和查询站点的处理

远程站点的维护算法如算法 2 所示。在初始化阶段, 远程站点 i 从协调者站点接收参数 (c_i, ψ, L_i, U_i) , 调用 $handleV(V_{i,0}, c_i, \psi, L_i, U_i)$ 初始化 SF_i ((1)、(2)行)。在正常处理阶段, 若(1)从协调者站点接收到 (L_i, U_i) , 或者(2)检测到新 $V_{i,t}$ 值, 且 $dist(V_{i,t}, c_i) \notin [l_i, u_i]$, 则调用 $handleV$ 函数更新过滤器 ((3)~(7)行); 若收到 $SEND$ 信号, 则向协调者站点发送 $V_{i,t}$ 。

算法 2 $remote()$

//远程站点 i

(1) 从查询站点接收到 (c_i, ψ, L_i, U_i) ;

(2) $handleV(V_{i,0}, c_i, \psi, L_i, U_i)$;

(3) LOOP

(4) IF (接收到 (L_i, U_i)) THEN

(5) $handleV(V_{i,t}, c_i, \psi, L_i, U_i)$;

(6) IF (接收到新 $V_{i,t}$, 且 $dist(V_{i,t}, c_i) \notin [l_i, u_i]$)

THEN

(7) $handleV(V_{i,t}, c_i, \psi, L_i, U_i)$;

向协调者站点发送 $V_{i,t}$;

(8) IF (接收到 $SEND$ 信号) THEN

(9) 向协调者站点发送 $V_{i,t}$;

算法 3 $coordinator(\psi)$

//初始化阶段, 仅仅执行一次

(1) 从所有远程站点接收各对象的当前位置 $\{V_{1,0}, V_{2,0}, \dots\}$;

(2) 生成集合 A , 含 k 个 $dist(V_{i,0}, q_0)$ 值最小的对象;

(3) $D_{inner} = \max(dist(V_{i,0}, q_0) | O_i \in A)$;

$D_{outer} = \min(dist(V_{i,0}, q_0) | O_i \in S - A)$;

(4) $B = (D_{inner} + D_{outer}) / 2$;

(5) FORALL (对象 $O_i, O_i \in A$)

$handleV(V_{i,0}, q_0, \psi, 0, B)$;

发送 $(q_0, \psi, 0, B)$ 给对象 O_i ;

(6) FORALL (对象 $O_i, O_i \notin A$)

$handleV(V_{i,0}, q_0, \psi, B, \infty)$;

发送 (q_0, ψ, B, ∞) 给对象 O_i ;

//正常处理阶段

- (7) WHILE (从远程站点 i 接收到数据 $V_{i,t}$)
 (8) $handleV(V_{i,t}, c_i, \psi, L_i, U_i)$;
 (9) IF (不能够根据定理 1~定理 5 计算查询结果) THEN
 (10) $rearrange()$;

算法 4 $rearrange()$

- (1) WHILE ($(A=\{O_i|R_i<k\}) \wedge (|A|<k)$)
 (2) 在集合 $S-A$ 中找到具有最小 d_i^- 值的对象 O_i ;
 (3) IF (协调者站点尚未获得对象 O_i 的最新值 $V_{i,t}$)
 THEN
 (4) 发送 $SEND$ 信号给 O_i , 从而接收 $V_{i,t}$;
 (5) $d_i^- = dist(V_{i,t}, q_i)$; $d_i^+ = dist(V_{i,t}, q_i)$;
 (6) $D_{inner} = \max(d_i^+ | O_i \in A)$; $D_{outer} = \min(d_i^- | O_i \in S-A)$;
 (7) $B = (D_{inner} + D_{outer}) / 2$;
 (8) FORALL(满足 $(d_i^- = d_i^+) \wedge (O_i \in A)$ 的对象 O_i)
 $handleV(V_{i,t}, q_i, \psi, 0, B)$;
 并发送 $(L_i, U_i) = (0, B)$ 给对象 O_i ;
 (9) FORALL(满足 $(d_i^- = d_i^+) \wedge (O_i \notin A)$ 的对象 O_i)
 $handleV(V_{i,t}, q_i, \psi, B, \infty)$;
 并发送 $(L_i, U_i) = (B, \infty)$ 给对象 O_i ;

协调者站点的处理算法(算法 3)含两阶段:初始化阶段和正常处理阶段。在初始化阶段,协调者站点从所有远程站点接收对象的位置 $\{V_{1,0}, V_{2,0}, \dots\}$; 创建集合 A , 含距离 q_0 最近的 k 个对象; 计算 B ; 计算过滤器参数; 并发送参数给远程站点(第(1)~(6)行)。在正常处理阶段, 每当从远程站点 i 接收到数据 $V_{i,t}$, 首先调用 $handleV$ 函数更新过滤器参数, 然后再根据定理 1~定理 5 判断是否能计算查询结果, 若不能, 则调用 $rearrange$ 函数重新调整。

算法 $rearrange$ 计算离协调者站点最近的 k 个对象。首先, 计算一个集合 A , 包含所有 $R_i < k$ 的对象, 这些对象均属于 k -最近邻集合。若 $|A| < k$, 则逐次在 $S-A$ 集合中找到一个对象, 获取其最新位置, 最终使得 $|A| = k$, 即: $A = kNN(q_i, k)$ 。由于每次获取 d_i^- 值最小的对象 ($S-A$ 中最可能离 q_i 最近的对象) 的最新值, $rearrange$ 函数仅需和少数对象进行通讯(第(1)~(5)

行)。在生成 A 之后, 计算 B , 重设部分满足 $(d_i^- = d_i^+)$ 的站点的过滤器参数。

3.4 性能分析

新方法的网络资源开销较小, 主要原因有三: 差异化的过滤器监控范围、轻量级的网络协议、使用松弛系数 ψ 。此外, 新方法的计算资源开销也很低。

3.4.1 差异化的过滤器监控范围

现有方法中, 往往众多过滤器共用少数几个监控范围。例如, RTP 方法中, 协调者站点维护全局参数 B , 部分对象(较少)的监控范围是 $[0, B]$; 其余对象(较多)均为 $[B, \infty]$ 。当距离小于 B 的对象过少时, 需要获取所有监控范围是 $[B, \infty]$ 的对象的当前位置, 从而增加网络开销。而新方法中, 各远程站点的监控范围 $[l_i, u_i]$ 各不相同, 取决于对象的位置。协调者站点在计算出包含 k -最近邻集合的直径 B 之后(算法 3 第(4)行, 算法 4 第(7)行), 对于较远的对象, 设置监控范围为 $[\frac{B + dist(V_{i,t}, q_i)}{2}, \infty]$ (算法 1 第(6)行), 而非 $[B, \infty]$ 。此项措施具有正负两方面的效应, 负面效应是远程对象更易于违反监控范围, 增大传输开销; 正面效应是运行 $rearrange$ 函数的网络开销大大降低。两者相比较, 正面效应所带来的益处远大于负面效应的消极作用。

3.4.2 轻量级的网络协议

该算法中, 远程站点与协调者站点间以轻量级协议进行通讯。共有四类报文:

(T1) 当 $dist(V_{i,t}, c_i) \notin [l_i, u_i]$ 时, 远程站点 i 发送 $V_{i,t}$ 给协调者站点(算法 2 的第(7)行);

(T2) 协调者站点发送 $SEND$ 信号给远程站点, 请求最新数据(算法 4 第(4)行);

(T3) 远程站点响应 $SEND$ 信号, 发送新数据给协调者站点(算法 2 第(9)行);

(T4) 协调者站点发送过滤器参数 (L_i, U_i) 给部分远程站点(算法 4 的第(8)、(9)行)。各个报文所含信息很少。

另外, 当协调者站点收到报文 T1 时, 调用 $han-$

dleV 函数更新本地过滤器；同时，远程站点也调用 handleV 函数更新远程过滤器，避免了再次通讯。

3.4.3 使用松弛系数 ψ

协调者站点将全局松弛系数 ψ 发送给所有远程站点，以辅助更新过滤器的监控范围(算法 3)。图 4 演示了使用与不使用 ψ 参数的两种情况。各圆点和曲线记录了对象与协调者站点之间距离的变化。初始阶段，监控范围是 $[l, \infty]$ 。图 4(a)未采用 ψ 参数(或者说 $\psi=0$)，因此当 $dist(V_{i,t}, q_i)$ 逐步减小乃至越过 l 时， l 的新值总是 $\frac{dist(V_{i,t}, q_i)+L}{2}$ 。而图 4(b)采用了 ψ 参数，当 l 值较为接近 L 时，直接降到 L ，从而降低了设置新门槛的频率。从图 4 可以看出，使用 ψ 参数之后，新范围的设置次数从 5 次降到 3 次。

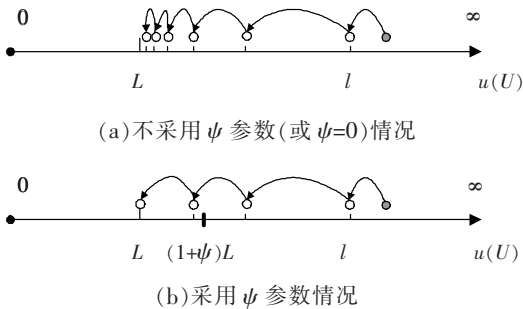


图 4 松弛系数 ψ 的使用
Fig.4 The usage of parameter ψ

3.4.4 计算资源开销较小

新方法消耗的计算资源也较小。远程站点在处理新数据时，仅需判断新值是否越过了门槛 $[l_i, u_i]$ ，并偶尔更新过滤器设置。协调者站点的计算开销主要有两处：判断能否基于当前过滤器参数设置计算查询结果，和调用 rearrange 函数。前者的瓶颈在于对 d_i^- 和 d_i^+ 值大小的判断，分别为 $\{d_i^-\}$ 和 $\{d_i^+\}$ 建立 B-树即可提高运行效率；后者的瓶颈是三个循环过程(算法 4 的第 (1)、(8)、(9)行)，由于嵌套次数较少，计算量也不高。

4 面向移动协调者站点的单查询处理方法

最近邻查询应用中的查询点可以不局限于静止

对象(例如城市地标建筑等)，还可以是一个移动物体(例如出租车、动物等)。文章的新方法能较好地解决这个问题。

在时间 t ，协调者站点不仅仅知道各个远程站点的过滤器设置 $(c_i, l_i, u_i, \psi, L_i, U_i)$ ，还知道自身位置 q_i 。据此，它可以估算对象 O_i 至自身的最小和最大距离 d_i^- 和 d_i^+ 。

定理 6 在时间 t ，协调者站点可估算对象 O_i 至协调者站点之间的距离范围是 $[d_i^-, d_i^+]$ ，且 d_i^- 和 d_i^+ 分别如公式(4)和(5)所示。

$$d_i^- = \begin{cases} l_i - dist(q_t, c_i) : dist(q_t, c_i) < l_i \\ 0 : (dist(q_t, c_i) \leq u_i) \wedge (dist(q_t, c_i) \geq l_i) \\ dist(q_t, c_i) - u_i : dist(q_t, c_i) > u_i \end{cases} \quad (4)$$

$$d_i^+ = dist(q_t, c_i) + u_i \quad (5)$$

证明 远程站点 i 在时间 t 的位置是 $V_{i,t}$ 。根据性质 1~性质 3，可知：

$$dist(V_{i,t}, q_t) \leq dist(q_t, c_i) + dist(V_{i,t}, c_i) \leq dist(q_t, c_i) + u_i = d_i^+$$

对于 d_i^- 而言，分三种情况考虑。

(1) 当 $dist(q_t, c_i) < l_i$ 时， $dist(V_{i,t}, q_t) \geq dist(V_{i,t}, c_i) - dist(q_t, c_i) \geq l_i - dist(q_t, c_i)$ ；

(2) 当 $dist(q_t, c_i) \in [l_i, u_i]$ 时， q_t 可能恰巧是 $V_{i,t}$ 的位置，据性质 2， $d_i^- = 0$ ；

(3) 当 $dist(q_t, c_i) > u_i$ 时， $dist(V_{i,t}, q_t) \geq dist(q_t, c_i) - dist(V_{i,t}, c_i) \geq dist(q_t, c_i) - u_i$ 。

得证。因此，依然可以利用定理 1~定理 5 计算查询结果。

例如，图 5 的左图和右图分别表示协调者站点北移前后对象 O_i 的 d_i^- 和 d_i^+ 的变化情况。阴影部分表

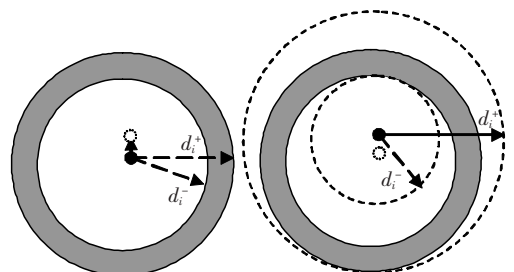


图 5 协调者站点移动时 d_i^- 和 d_i^+ 的变化情况

Fig.5 The change of d_i^- and d_i^+ as the coordinator site moves

示对象 O_i 的可能位置。右图虚线圆环表示夹在以 d_i^+ 和 d_i^- 为直径的两个圆间的环。

值得一提的是,处理移动协调者站点情况时需要稍微修改算法框架以维护过滤器 SF 的 c 字段。在接收到新值 $V_{i,t}$ 后,协调者站点并不马上更新 c 字段,以与远程站点的过滤器设置保持一致。在 *rearrange* 函数(算法 4)的第(8)、(9)行,协调者站点在调用 *handleV* ($V_{i,t}, q_t, \psi, 0, B$) 或 *handleV* ($V_{i,t}, q_t, \psi, B, \infty$) 更新过滤器参数后,向部分远程站点发送 (q_t, L_i, U_i) , 而非 (L_i, U_i) 。远程站点接收到 (q_t, L_i, U_i) 之后,调用 *handleV* ($V_{i,t}, q_t, \psi, L_i, U_i$) 更新本地过滤器 SF_i (算法 2 的第(4)、(5)行)。由于篇幅限制,在此不重复列出 *remote* 函数和 *rearrange* 函数。

5 多最近邻查询的处理

对于多最近邻查询情况,现有工作往往独立处理各查询,较少考虑到信息共享因素。文中的新方法能统一处理所有最近邻查询。假设协调者站点注册了 m 个最近邻查询,各查询的最近邻数目参数分别是 k_1, k_2, \dots, k_m 。过滤器的处理函数 *handleV* 和远程站点的处理算法 *remote* 无需修正。如前所述,协调者站点以两阶段方式运行 *coordinator*: 初始化阶段和正常处理阶段。在此,初始化阶段(算法 3 的第(1)~(6)行)由算法 5(*coordinator_init*) 替代;正常处理阶段则调用另一版本的 *rearrange* 算法(算法 6)。

算法 5 在设置过滤器参数时综合考虑了注册的查询和各对象的当前位置。例如,图 6 显示了 12 个远程站点的位置分布,协调者站点上注册了 2 个最近邻查询, $k_1=4, k_2=6$ 。则 B_1 和 B_2 的值分别如图 6 所示。离 q 最近的 4 个站点的 $[L, U]=[0, B_1]$, 中间 2 个站点的 $[L, U]=[B_1, B_2]$, 其余站点的 $[L, U]=[B_2, \infty]$ 。各远程站点根据相应的 $[L, U]$ 值设置过滤器参数。

改进后的 *rearrange* 算法如算法 6 所示。该算法从距离协调者站点最近的对象开始查找,获取部分站点的最新值,直到所有查询条件(定理 1~定理 5)均满足。

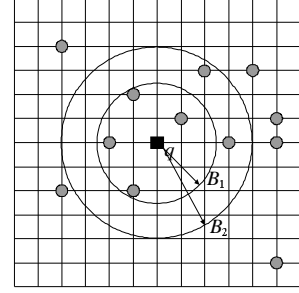


图 6 注册了 2 个查询

Fig.6 When two queries are registered

算法 5 *coordinator_init*()

- (1) 从所有远程站点接收对象的当前值 $\{V_{1,0}, V_{2,0}, \dots\}$;
- (2) $B_0=0$; 集合 $A_0=0$; A_1, A_2, \dots, A_m 分别含 k_1, k_2, \dots, k_m 个 $\text{dist}(V_{i,0}, q_0)$ 值最小的对象;
在此假定所有查询均已排序,满足:
 $k_1 \leq k_2 \leq \dots \leq k_m$ 。

(3) FOR $j=1$ TO m

- (4) $D_{j,inner} = \max(\text{dist}(V_{i,0}, q_0) | O_i \in A_j)$;
 $D_{j,outer} = \min(\text{dist}(V_{i,0}, q_0) | O_i \in S - A_j)$;
- (5) $B_j = (D_{j,inner} + D_{j,outer}) / 2$;
- (6) FORALL (对象 $O_i, O_i \in A_j - A_{j-1}$)
- (7) *handleV* ($V_{i,0}, q_0, \psi, B_{j-1}, B_j$);
发送 $(q_0, \psi, B_{j-1}, B_j)$ 给对象 O_i ;

算法 6 *rearrange*()

//多查询版本的 *rearrange* 算法

- (1) FOR $j=1$ TO m
- (2) WHILE $((A_j = \{O_i | R_i < k_j\}) \wedge (|A_j| < k_j))$
- (3) 在集合 $S - A_j$ 中查找具有最小 d_i^- 值的对象 O_i ;
- (4) IF (查询站点尚未收到对象 O_i 的最新值 $V_{i,t}$) THEN
- (5) 发送 SEND 信号给 O_i , 从而接收 $V_{i,t}$;
- (6) $d_i^- = \text{dist}(V_{i,t}, q_t)$; $d_i^+ = \text{dist}(V_{i,t}, q_t)$;
- (7) $D_{j,inner} = \max(d_i^+ | O_i \in A_j)$;
 $D_{j,outer} = \min(d_i^- | O_i \in S - A_j)$;
- (8) $B_j = (D_{j,inner} + D_{j,outer}) / 2$;
- (9) FORALL (满足 $(d_i^- = d_i^+) \wedge (O_i \in A_j - A_{j-1})$ 的对象 O_i)

handleV($V_{i,t}, q_i, \psi, B_{j-1}, B_j$);
 发送(q_i, B_{j-1}, B_j)给对象 O_i ;

6 实验

该节验证新方法的性能。分布式应用的最关键目标之一是降低网络传输开销。首先产生一个模拟数据集，测试新方法处理静止协调者站点场景的网络开销；然后再测试新方法在移动协调者站点和多查询场景下的网络开销；最后，报告新方法在一个真实数据集上的实验结果。实验代码全部由 C 语言编写，并运行于一台 Pentium IV 主机上。

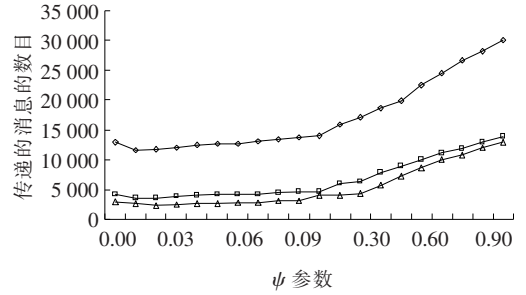
6.1 面向静止协调者站点

这里生成一个数据集，模拟 100 辆出租车一天的运行路线。首先，每辆车被随机给定一个初始坐标 (x_0, y_0) ，一个初始行车方向 a 和初始速度 b ；其次，每隔一秒钟产生一个新坐标 (x_t, y_t) ，满足： $(x_t, y_t) = (x_{t-1} + b * \cos(a), y_{t-1} + b * \sin(a))$ 。每隔一分钟，改变车辆的行车方向 a 和速度 b 。其中， a 和 b 分别在 $[0, 2\pi]$ 和 $[5, 15]$ 之间随机抽取。因此，100 辆车一天共产生 $100 * (60 * 60 * 24) = 8\ 640\ 000$ 个坐标记录。

如 3.4.3 节所述，松弛系数 ψ 是新方法的一个重要参数，被用于辅助设置过滤器的范围。在协调者站点单独运行三个最近邻查询 $r-kNN((0,0), 15, 0)$ 、 $r-kNN((0,0), 15, 2)$ 和 $r-kNN((0,0), 15, 5)$ 。 ψ 参数的值从 0.01 到 1.0。图 7 报告了各查询在不同 ψ 参数下的网络传输开销。总的来说，当 ψ 参数增大时，所产生的约束范围 $[l, u]$ 较小，导致网络传输开销增加；然而若 $\psi=0$ ，由于范围 $[l, u]$ 被远程对象的新位置频繁突破（如图 4 所示），也会导致传输开销增加（高出最低点 10%~15%）。综合考虑，当 ψ 值略大于 0 时，传输开销最小。文章的后续实验均采用 $\psi=0.02$ 。

新方法的计算资源开销也非常小。图 8 显示了协调者站点分别处理三个查询时的计算开销。横轴表示各个时刻，纵轴表示协调者站点在 2 小时内总的计算时间（扣除数据传输、网络延时等）。可以看

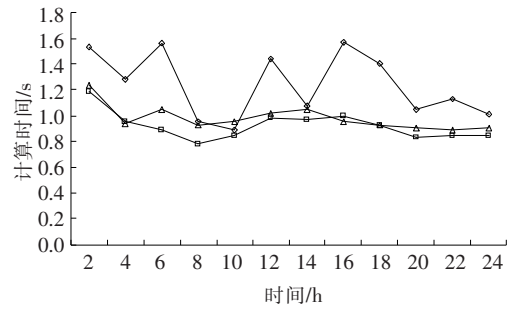
出，尽管在不同时段，计算时间有所波动，但总的来说，时间开销均非常小，适合处理实时应用场景。



—◇— $r-kNN((0,0), 15, 0)$ —□— $r-kNN((0,0), 15, 2)$ —△— $r-kNN((0,0), 15, 5)$

图 7 不同 ψ 参数设置下的传输开销

Fig.7 Transferring cost with different ψ

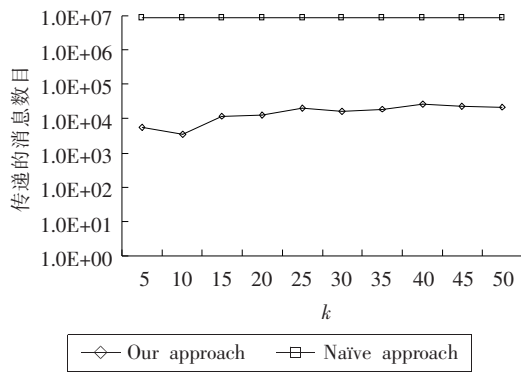


—◇— $r-kNN((0,0), 15, 0)$ —□— $r-kNN((0,0), 15, 2)$ —△— $r-kNN((0,0), 15, 5)$

图 8 处理最近邻查询的时间开销

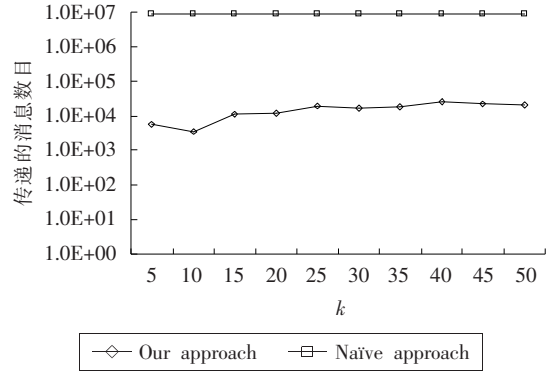
Fig.8 Time cost of processing NN queries

文章支持五种近似最近邻查询。图 9 显示了新方法处理绝对数值容许度最近邻查询、相对数值容许度最近邻查询、纳伪容许度最近邻查询和排名容许度最近邻查询等四种近似查询的传输开销（由于拒真容许度查询等同于纳伪容许度查询，因此无需单列）。在 Naïve approach 中，各远程站点每秒钟向协调者站点报告当前位置，以供其处理查询。易知，在各种条件下，新方法(Our approach)大大优于 Naïve approach。针对纳伪容许度查询和排名容许度查询，分别与 FT-NRP 方法、RTP 方法进行比较，如图 9(c)和图 9(d)所示。可以看出，新方法的性能大大优于这两个方法。



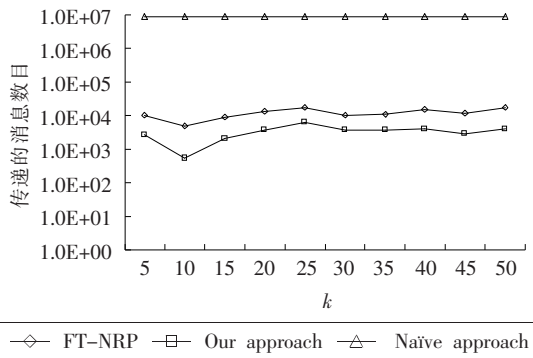
(a)处理 $a-kNN((0,0),k,50)$

(a)Processing $a-kNN((0,0),k,50)$



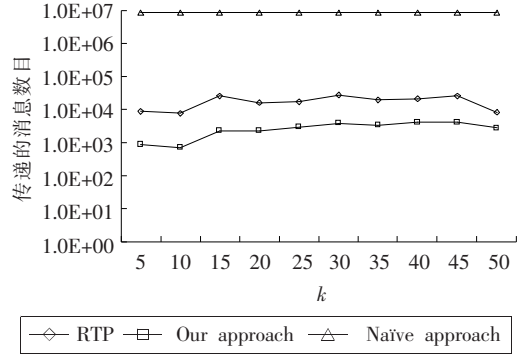
(b)处理 $v-kNN((0,0),k,0.1)$

(b)Processing $v-kNN((0,0),k,0.1)$



(c)处理 $p-kNN((0,0),k,0.2)$

(c)Processing $p-kNN((0,0),k,0.2)$



(d)处理 $r-kNN((0,0),k,5)$

(d)Processing $r-kNN((0,0),k,5)$

图9 处理各种近似最近邻查询的传输开销

Fig.9 Transferring cost in processing various nearest neighbor queries

新方法中, 远程站点与协调者站点间以四类报文进行通讯,T1、T2、T3 和 T4(第 3.4.2 节)。图 10 描述了在处理查询 $p-kNN((0,0),k,0.2)$ 和 $r-kNN((0,0),k,2)$ 时各类报文的传输数目情况。T2 报文和 T3 报文的数目相同;T2 报文的数目略少于 T4 报文传输的数目;大多数情况下,T1 报文的数目最少。

6.2 面向移动协调者站点

新方法同样支持移动协调者站点的场景。假定其中一辆出租车(标记为 q)是协调者站点,运行排名容许度最近邻查询 $r-kNN(q,k,2)$;其余 99 辆为远程站点。图 11 显示了处理该查询的网络传输开销。相比较静止协调者站点的场景,由于协调者站点持续移动,其最近邻集合的变化更为剧烈,导致更多的网络开销。但是,其开销仍远远小于 Naïve approach。

6.3 面向多最近邻查询

新算法支持多最近邻查询情况。假定协调者站点静止,在上面注册两个最近邻查询 $r-kNN((0,0),5,e)$ 和 $p-kNN((0,0),25,e*0.04)$ 。图 12 分别列举了单独运行这两个查询,以及同时运行这两个查询所耗费的网络资源。可以看出,由于统一处理多个查询,同时运行两个查询所耗费的网络开销要比分别运行两个查询的网络开销之和小得多。

6.4 基于真实数据集

作者还在真实数据集上验证新方法的性能,该集合^[10]包含一天(1998 年 7 月 15 日)内访问世界杯'98 官方网站的所有点击记录。共有 30 个服务器站点为世界杯'98 官方网站服务。假设每隔一秒钟,各站点统计过去一分钟内处理的点击数,并发送给

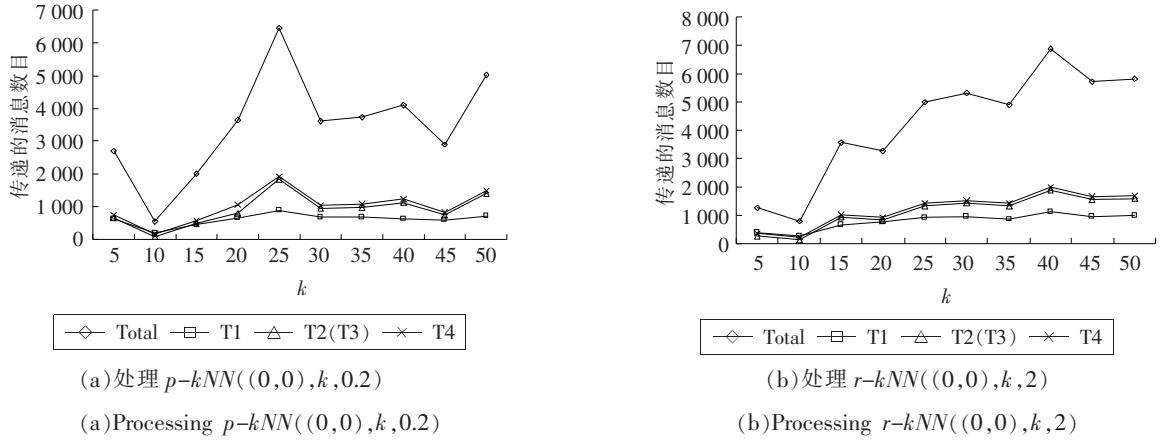


图 10 新方法的协议分析

Fig.10 Protocol analysis of our approach

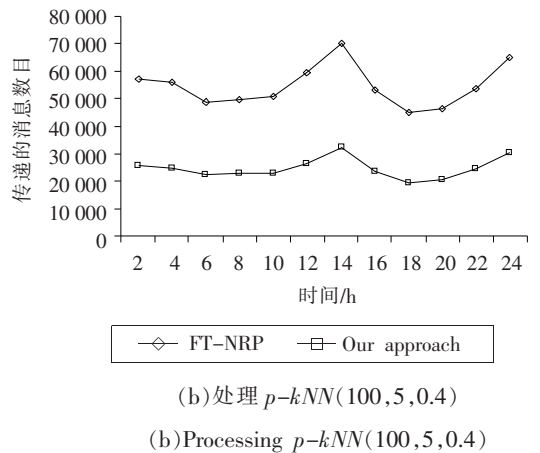
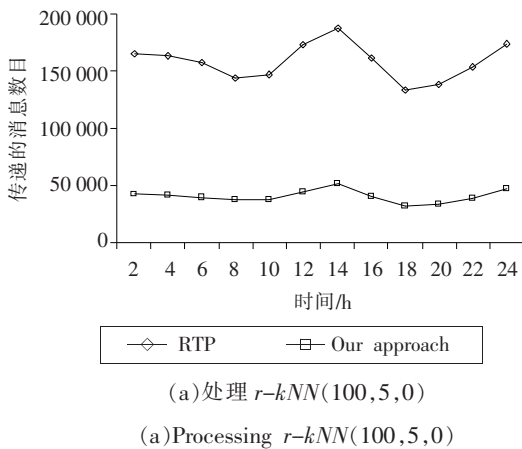
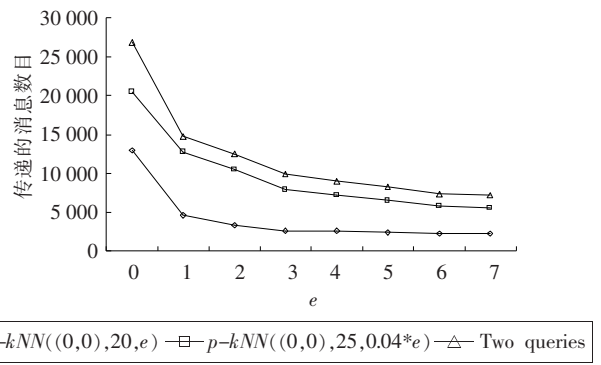
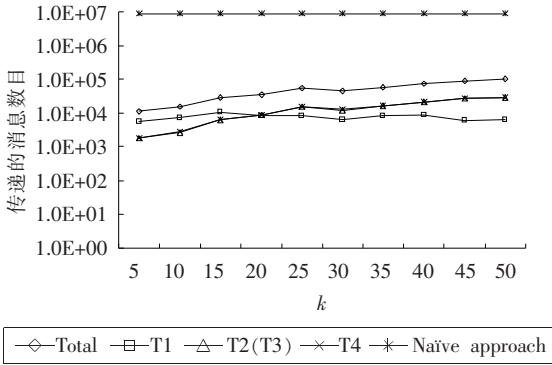


图 13 在真实数据集上处理最近邻查询

Fig.13 Processing nearest neighbors queries upon real dataset

一个中心主机。图 13 显示了运行查询 r - k NN (100,5,0), p - k NN(100,5,0.4)的结果。横轴表示各个时刻;纵轴表示在过去 2 小时内各方法传输的消息数目。可以看出,在各种情况下,新方法均优于 RTP 方法和 FT-NRP 方法。

7 相关工作

近年来,针对新型应用中的高速数据流的研究工作进展很快^[1-3],出现了一些连续监控(continuous monitoring)方法,它们连续监控新产生的数据,并实时输出结果,例如 top- k 监控^[6],简单聚集函数查询^[11],相异元素估算^[12]等。

最近邻问题得到了广泛的研究。传统方法^[5,13,14]一般假设数据存于磁盘上,且可多次访问,主要的方法是创建和维护各种索引结构以优化查询,但是这类方法不适用于处理高速数据。新近的研究成果^[4,7-9,15]则能够处理高速数据。Iwerks 等人^[15]提出的 CW 方法主要针对 PRO 对象(point kinematic object);Koudas 等人^[4]提出的 DISC 方法支持近似 kNN 查询。但是,这两个算法均不是面向分布式环境,没有考虑数据传输开销。Cheng 等人提出的 RTP 方法和 FT-NRP 方法能支持分布式环境,其目的是节约网络传输开销^[7],但是该方法不支持移动协调者站点和多查询处理情况。Mouratidis 等人提出一种基于门槛的算法,能处理移动协调者站点情况,但是该方法并未考虑近似最近邻查询^[8]。Jin 等人提出的方法支持部分近似最近邻查询^[9]。

最近邻研究的其他热点问题还包括:reverse kNN、基于道路拓扑的查询和聚集查询等。一个 reverse kNN 查询 $RNN(k,q)$ 要求获取所有将 q 视为其 k 个最近邻之一的对象^[16]。Kolahdouzan 等人研究了在预定义的路面拓扑图上做最近邻查询的问题^[17]。Papadias 等人研究了在空间时间数据库上的聚集最近邻查询问题^[18]。

8 结论

降低网络带宽耗费是分布式实时监控应用的关

键目标之一。文章研究分布式环境下的最近邻查询问题。目前,一些研究人员通过为移动对象建立各种索引以降低查询的响应时间,但是传输开销较大;还有部分研究人员设计基于门槛的方法,以降低传输开销。文章提出了一个新方法,相比现有方法:(1)支持多种近似最近邻查询;(2)支持移动协调者站点;(3)支持多查询处理;(4)更节约网络资源。理论分析和实验结果均表明,新方法优于现有方法。

References:

- [1] Babcock B, Babu S, Datar M, et al. Models and issues in data stream systems[C]//Papa L. Proceedings of the Twenty-first ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, PODS 2002, Madison, Wisconsin, USA. New York: ACM Press, 2002: 1-16.
- [2] Golab L, Özsu M T. Issues in data stream management[J]. SIGMOD Record, 2003,32(2):5-14.
- [3] Jin C, Qian W, Zhou A. Analysis and management of streaming data: a survey[J]. Journal of Software, 2004,15(8):1172-1181.
- [4] Koudas N, Ooi B, Tan K, et al. Approximate NN queries on streams with guaranteed error/performance bounds[C]//Nascimento M A, Özsu M T, Kossmann D, et al. Proceedings of the Thirtieth International Conference on Very Large Data Bases, VLDB 2004, Toronto, Canada. San Fransisco: Morgan Kaufmann, 2004:804-815.
- [5] Yu C, Ooi B, Tan K L, et al. Indexing the distance: an efficient method to kNN processing [C]//Aperis P, Atzeni P, Ceri S, et al. Proceedings of the 27th International Conference on Very Large Data Bases, VLDB 2001, Roma, Italy. San Fransisco: Morgan Kaufmann, 2001:421-430.
- [6] Babcock B, Olston C. Distributed top- k monitoring[C]//Halevy A Y, Ives Z G, Doan A. Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, ACM SIGMOD 2003, San Diego, California, USA. New York: ACM Press, 2003:28-39.

- [7] Cheng R, Kao B, Prabhakar S, et al. Adaptive stream filters for entity-based queries with non-value tolerance[C]// Böhm K, Jensen C S, Haas L M, et al. Proceedings of the 31st International Conference on Very Large Data Bases, VLDB 2005, Trondheim, Norway. New York: ACM Press, 2005:37-48.
- [8] Mouratidis K, Papadias D, Bakiras S, et al. A threshold-based algorithm for continuous monitoring of k nearest neighbors[J]. IEEE Transactions on Knowledge and Data Engineering, 2005,17(11):1451-1464.
- [9] Jin C, Guo W. Efficiently monitoring nearest neighbors to a moving object[C]//Alhajj R, Gao H, Li X. Proceedings of the 3rd Advanced Data Mining and Applications, ADMA 2007, Harbin, China. Berlin: Springer, 2007.
- [10] World cup '98 dataset[EB/OL]. <http://ita.ee.lbl.gov>.
- [11] Olston C, Jiang J, Widom J. Adaptive filters for continuous queries over distributed data streams[C]//Halevy A Y, Ives Z G, Doan A. Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, ACM SIGMOD 2003, San Diego, California, USA. New York: ACM Press, 2003:563-574.
- [12] Das A, Ganguly S, Garofalakis M, et al. Distributed set-expression cardinality estimation[C]//Nascimento M A, Özsu M T, Kossmann D, et al. Proceedings of the Thirtieth International Conference on Very Large Data Bases, VLDB 2004, Toronto, Canada. San Francisco: Morgan Kaufmann, 2004:312-323.
- [13] Hjaltason G, Samet H. Ranking in spatial databases[C]// Egenhofer M J, Herring J R. Advances in Spatial Databases, the 4th International Symposium, SSD 1995, Portland, Maine, USA. Berlin: Springer, 1995:83-95.
- [14] Weber R, Schek H J, Blott S. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces[C]//Gupta A, Shmueli O, Widom J. Proceedings of the 24th International Conference on Very Large Data Bases, VLDB 1998, New York, USA. San Francisco: Morgan Kaufmann, 1998:194-205.
- [15] Iwerks G, Samet H, Smith K. Continuous k -nearest neighbor queries for continuously moving points with updates[C]//Freytag J C, Lockemann P C, Abiteboul S, et al. Proceedings of the 29th International Conference on Very Large Data Bases, VLDB 2003, Berlin, Germany. San Francisco: Morgan Kaufmann, 2003:512-523.
- [16] Yiu M L, Papadias D, Mamoulis N, et al. Reverse nearest neighbors in large graphs[J]. IEEE Transaction on Knowledge and Data Engineering, 2006,18(4):540-553.
- [17] Kolahdouzan M R, Shahabi C. Alternative solutions for continuous k nearest neighbor queries in spatial network databases[J]. GeoInformatica, 2005,9(4):321-341.
- [18] Papadias D, Tao Y, Mouratidis K, et al. Aggregate nearest neighbor queries in spatial databases[J]. ACM Transactions on Database Systems, 2005,30(2):529-576.

附中文参考文献:

- [3] 金澈清, 钱卫宁, 周傲英. 流数据分析与管理综述[J]. 软件学报, 2004,15(8):1172-1181.



金澈清(1977-),男,浙江文成人,博士,1999年和2002年于浙江大学分别获得计算机专业工学学士学位和工学硕士学位,2005年于复旦大学获得计算机软件与理论专业博士学位,目前是华东理工大学计算机系讲师,主要研究领域为数据挖掘和数据流处理。

JIN Cheqing was born in 1977. He is currently an assistant professor in Computer Science at East China University of Science and Technology. He received his Bachelor and Master degrees in Computer Science from Zhejiang University in Hangzhou, P.R. China in 1999 and 2002 respectively, and the PhD degree from Fudan University, Shanghai, P.R. China in 2005. He worked as a research assistant at E-Business Technology Institute, the Hong Kong University from December 2003 to May 2004. His current research interests include data mining and data stream.



崇志宏(1969-),男,江苏盐城人,博士,东南大学计算机科学与工程学院讲师,主要研究方向:数据挖掘,流数据管理和 P2P、传感器网络环境下的数据管理。

CHONG Zhihong was born in 1969. He received his BS in Computer Science from Nanjing Meteorological Institute and MS degree in Economics from Finance Institute of Ministry of Finance People's Republic of China in 1991 and 1999 respectively. He received his PhD degree from Fudan University in 2006. He is currently a lecture at Southeast University, China. His research interests include data mining, data streams and data management in P2P and sensor network environment. He has published several research papers in these areas in major international conferences and reputable journals.



周傲英(1965-),男,安徽郎溪人,博士,中国计算机学会高级会员,1985年和1988年于四川大学分别获得计算机专业工学学士学位和工学硕士学位,1993年于复旦大学获得计算机软件与理论专业博士学位,目前是复旦大学计算机系教授。他曾经作为许多国际学术会议的程序会主席或委员,例如 WWW, SIGMOD, VLDB, EDBT, ICDCS, ER, DASFAA, PAKDD, WAIM。他的文章已经发表在 ACMSIGMOD, VLDB, ICDE 及其他国际期刊上。主要研究领域是数据挖掘和知识发现,XML 数据管理,Web 挖掘和搜索,数据流分析处理,P2P 计算等。

ZHOU Aoying was born in 1965. He is currently a senior membership of China Computer Federation, a professor in Computer Science at Fudan University, Shanghai, P.R. China. He received his Bachelor and Master degrees in Computer Science from Sichuan University in Chengdu, Sichuan, P.R. China in 1985 and 1988, respectively, and PhD degree from Fudan University in 1993. He served as the member or chair of program committee for many international conferences such as WWW, SIGMOD, VLDB, EDBT, ICDCS, ER, DASFAA, PAKDD, WAIM, etc. His papers have been published in ACMSIGMOD, VLDB, ICDE, and several other international journals. His research interests include data mining and knowledge discovery, XML data management, Web mining and searching, data stream analysis and processing, peer-to-peer computing.