doi:10.3969/j.issn.1000-436x.2014.03.023

断接下移动终端的简单查询算法研究

梁茹冰[1]，刘琼[2]
(1. 华南农业大学 理学院，广东 广州 510642；2. 华南理工大学 软件学院，广东 广州 510006)

摘 要：提出了断接下移动终端简单查询算法 SQPID，该算法通过合并与裁剪操作构建综合相关语义缓存项，且合并过程不涉及间接相关性判断，从而简化了以往算法的处理过程，提高了近似查询结果的导出速度。实验表明，SQPID 算法在查询响应时间和精确度方面都更好地满足了用户的需求。
关键词：语义缓存；断接；简单查询算法；综合相关语义缓存项
中图分类号：TP311 **文献标识码**：A **文章编号**：1000-436X(2014)03-0201-07

Research on simple query algorithm during mobile terminal disconnection

LIANG Ru-bing[1], LIU Qiong[2]
(1. College of Science, South China Agricultural University, Guangzhou 510642, China;
2. School of Software Engineering, South China University of Technology, Guangzhou 510006, China)

Abstract: A simple query processing algorithm SQPID (simple query processing in disconnection), was proposed. Main idea of SQPID is to build integrated related semantic cache item about current query through combining and clipping operations. In order to simplify the procedure, SQPID's combine does not involve indirect relativity estimation. The results of simulated experiments show that SQPID's query response time and degree of accuracy better meet client's requirements than QPID algorithm.
Key words: semantic caching; disconnection; simple query algorithm; integrated related semantic cache item

1 引言

在无线移动计算环境中，终端为了节约有限的带宽资源和电池能量，会经常与网络连接断开，而在另一时间再与网络连接以获取数据。终端当前如果处于断接状态时，提交的查询请求无法获得精确解。往往可以利用本地缓存数据信息来提供近似答案，从而在一定程度上满足用户的需要。这种满足是非常必要的，相比直接告诉用户无法查询，近似查询的方式则更加友好。事实上，用户也并不希望系统随时随地返回精确结果，这样将会大大增加流量及流量消耗。在一些信号差或无信号的网络计算环境中，近似查询结果也可以让人接受。本文研究在频繁断接情况下，利用本地语义缓存如何有效地得到近似查询结果的方法。

2 相关研究

2.1 语义缓存

语义缓存领域的研究主要表现在一致性维护方法、缓存理论、缓存替换方法和缓存预取策略这 4 个方面。近 3 年来，在缓存理论[1]和一致性维护[2]方面做了些研究工作。在语义缓存理论研究方面，Dar 等人[3]最先提出语义缓存结构，在单个关系表的基础上研究了语义缓存查询处理过程。在语义缓存形式化方面，Ren 等人[4,5]提出了缓存项的元组形式，并进一步给出判断当前查询与语义缓存匹配的

收稿日期：2012-09-04；**修回日期**：2014-02-16
基金项目：国家重点基金研究发展计划（“973”计划）基金资助项目(2007CB07100, 2007CB07106)；国家自然科学基金资助项目(61171141)
Foundation Items: The National Basic Research Program of China (973 Program) (2007CB07100, 2007CB07106); The National Natural Science Foundation of China (61171141)

3 个判定定理，并提出在 5 种相交情况下，语义缓存查询裁剪算法。国内研究成果代表的有吴婷婷等人[6,7]对查询导出方法的研究，并提出查询与缓存项匹配的判定定理，进而在所提理论基础上实现了一个语义缓存查询处理模型。另外，还有蔡建宇等人[8]对聚集查询的语义缓存进行了研究，构造了服务于海量数据库的语义缓存系统，并进一步研究提出缓存管理、查询处理和一致性维护等关键技术。在前人工作基础上，提出 SMC-架构即由远程服务方、移动支持站点和移动终端组成的 3 层架构[2]。以 SMC-架构为运行平台，移动支持站点在服务方和终端之间起到中介的作用，可以协助移动终端进行缓存管理和维护。具体方法是采用缓存项索引表和位序列法生成简单的缓存失效报告，利用广播通道发送给移动终端；终端读取报告并从中提取出与本地缓存中匹配的部分进行一致性标记和维护，为了更准确地标定失效缓存项，提出了粒度细化方法。

2.2 断接近似查询

这方面的研究成果代表性的是吴婷婷等人[9]提出的终端断接下语义缓存查询处理(QPID, query processing in disconnection)算法。根据当前查询内容，QPID 算法先找出本地缓存中与其相关的缓存项，再合并处理多个相关项并从中导出查询的近似结果。不过 QPID 算法需要进行较多间接缓存相关项判断和合并，实现较为复杂，且响应时间较长。陈良刚等人[10]提出并设计概括数据库进行断接位置相关查询，提交查询请求后，通过访问位置概念层次树返回不同程度的近似查询结果。何文麟等人[11]提出索引多维范围查询的多叉树结构，这一结构可达到优化近似连续范围查询的目的，但算法不适用于断接情况。

由上可见，已有的研究成果提供了一定的理论基础，当前断接近似查询存在的主要问题是相关性判断过程复杂以及响应时间较长。文中应用语义缓存方法，提出断接下移动终端简单查询算法(SQPID)，该算法通过简化相关项判断过程，达到加快断接查询结果导出速度的目的。

3 断接情况下的简单查询算法

3.1 定义及相关说明

定义 1 可缓存查询是指某查询条件语句其结构上形如“SELECT-FROM-WHERE”，其中，SELECT 部分涉及的投影属性集来自同一个关系表，FROM 部分只能涉及一个关系表，WHERE 部分是查询条件子句，这是由多个“小于”、“大于”和“等于”表达式组成的合取式。

类似文献[4]，可缓存查询表达式中不考虑形如“$X\neq a$”的条件比较表达式，而是采用等价形式“$(X<a)\vee(X>a)$”来代替“$X\neq a$”，所以表达能力没有受到影响。移动终端提交的查询，只有满足可缓存查询条件，才将缓存描述和查询结果保存在本地缓存中；否则，将查询请求发送至远程服务器处理，结果也不缓存。这样处理是因为不满足可缓存查询条件的查询表达式形式上相对复杂，出现概率低，再次出现的可能性更小，对其缓存并维护一致性的意义不大。

定义 2 终端提交的查询请求形式化表示为 4 元组 $Q=<Q_R, Q_A, Q_P, Q_C>$。其中，$Q_R\in D$，R_i 为第 i 个关系表，数据库 $D=\bigcup_{i=1}^{n}\{R_i\}$ 由 n 个关系表组成；Q_A 为查询的投影属性集；Q_P 为查询条件谓词；$Q_C=\pi_{Q_A}\sigma_{Q_P}(Q_R)$ 为查询结果的缓存数据块。

定义 3 终端查询对应的语义缓存索引项 S_Q 可形式化表示为 5 元组 S_Q= <S_R，S_A，S_P，S_C，S_T>。其中，S_R 和 S_A 分别描述语义片段对应的二维关系表和查询结果集中涉及的属性集合；S_P 是约束条件；S_C 是 S_Q 的实际内容；S_T 是语义片段的当前时间标记，用于进行一致性验证。语义缓存索引项数据存储在移动终端中，可以对终端本地缓存中的数据内容及存储位置进行语义描述。

为了进一步说明查询的完备性、正确性和等价性等概念，设 D 为服务器数据库，D_S 为移动终端本地语义缓存数据库，显然有关系 $D_s\subseteq D$。对于终端提交的可缓存查询 Q，$A=\{a_1,a_2,\cdots,a_n\}$ 表示由 D 返回的查询结果元组集合，$B=\{b_1,b_2,\cdots,b_m\}$ 表示由 D_S 返回的查询结果元组集合。

定义 4 查询的完备性指查询由 D 返回的结果至少是由 D_S 返回结果的一部分，判断条件为

$$(\forall a\in A, a\in B)\vee(\neg\exists a\in A, a\notin B)$$

例如，$A=\{1,3,5\}$，$B=\{1,2,3,5\}$，亦即集合 B 包含集合 A，这里 A 表示数据库服务器返回的查询结果，B 表示终端语义缓存数据库返回的查询结果。

定义 5 查询的正确性指查询在 D_S 上的结果至少是在 D 上结果的一部分，判断条件为 $\forall b\in B, b\in A$，

即 $B \subseteq A$，并且允许 $\exists a \in A, a \notin B$，即满足正确性但不满足完备性。例如，$A=\{1,3,6\}, B=\{1,6\}$，其中，$A$、$B$ 含义同定义 4。

定义 6　如果查询完备性和正确性至少有一个不满足时，称该查询为不精确查询，反之则称为精确查询，如表 1 所示。

表 1　查询的完备性、正确性和精确性

完备性	正确性	精确性
False	False	False
False	True	False
True	False	False
True	True	True

移动终端处于断接情况时，利用本地语义缓存的理想情况是能够获得既正确又完备的精确查询结果，不过通常得到的是正确但不完备的近似查询结果。因此，需要对近似结果的导出方法进行研究，并尽可能做到简化处理。

定义 7　查询的等价性亦可称为查询的精确性，等价查询指同时满足正确性和完备性的查询类型，也就是说，在断接或弱连接情况下，等价查询表现为由缓存返回的结果与数据库服务器返回的结果相同。

定义 8　移动终端中与当前查询相关，并且有可能包含查询近似结果的缓存项称为相关语义缓存索引项。利用它可以对断接查询提供正确不完备的近似结果。对于查询 Q=<Q_R, Q_A, Q_P, Q_C>，各语义缓存项如果满足如下 3 个条件即可认为是与该查询相关的索引项：1) Q_R=S_R；2) $(Q_A \cup Q_{PA}) \cap S_A \neq \Phi$；3) $Q_P \wedge S_P \neq \Phi$。通常，相关语义缓存索引项不止一个，此时需要将多个索引项内容进行合并，构建综合相关语义缓存项，再执行查询。

定义 9　合并多个当前查询的相关语义缓存索引项形成的数据块称为综合相关语义缓存项。具体构建方法见 2.2 节。

定义 10　断接下的查询结果 $X=\{x_1, x_2, \cdots, x_m\}$ 与精确解 $A=\{a_1, a_2, \cdots, a_n\}$ 的接近程度称为断接下查询的近似度，近似度可通过 $F(X)=\sum_{i=1}^{m}\sum_{j=1}^{p} x_{ij} / \sum_{i=1}^{n}\sum_{k=1}^{q} a_{ik}$ 来计算，其中 $x_{ij} \in \{0,1\}$，0 表示该属性列值为空，1 表示该属性列值不为空，精确解中各行各列的属性值均不为空时 $a_{ik}=1$，易见 $0 \leqslant F(X) \leqslant 1$。

不同于文献[9]中的处理方法，在本文中属性集 S_A 包含的内容由 3 部分组成，分别是：查询条件语句中涉及到的属性集、查询的投影属性集和关系表主键。也就是说，文中算法中的 S_A 内容包含了文献[9]中所提的扩展属性集和投影属性集。这样处理的目的是为了对间接相关语义缓存项的判定、转换和合并的时间进行缩减，对相关语义缓存索引项的判定和合并过程进行简化，从而达到加快近似查询结果导出速度的目的，详见第 3.2 节和第 3.3 节介绍。

3.2　综合相关语义缓存项构建

综合相关语义缓存项的构建过程由 2 个步骤组成，即合并操作和裁剪操作。

3.2.1　合并操作

合并是将多个相关语义缓存索引项进行内容整合，结果是原关系表内容的一部分，处理过程描述如下。

令 $S_1=<S_{R1}, S_{A1}, S_{P1}, S_{C1}>$，$S_2=<S_{R2}, S_{A2}, S_{P2}, S_{C2}>$，…，$S_n=<S_{Rn}, S_{An}, S_{Pn}, S_{Cn}>$ 是与当前查询 Q 相关的 n 个语义缓存索引项，合并后的语义缓存项为 4 元组表示为 $S_m=<S_{Rm}, S_{Am}, S_{Pm}, S_{Cm}>$。其中，$S_{Rm}=S_{R1}=S_{R2}=\ldots=S_{Rn}$，$S_{Am}=\bigcup_{i=1}^{n} S_{Ai}$，$S_{Pm}=\vee_{i=1}^{n} S_{Pi}$，$S_{Cm}$ 是 S_m 的实际内容。用 NULL 字符串填充 S_{Cm} 中元组的空缺属性的值。

3.2.2　裁剪操作

合并操作完成后，接着要进一步对 S_m 进行属性列裁剪，方法是将 S_{Am} 中与当前查询 Q 无关的属性列删掉，处理过程描述如下。

令合并后语义缓存项 $S_m=<S_{Rm}, S_{Am}, S_{Pm}, S_{Cm}>$，当前查询为 $Q=<Q_R, Q_A, Q_P, Q_C>$，Q_A 为查询涉及的属性集合，Q_{PA} 为查询条件中包含的属性集合，Q_{RK} 为关系表主键。对 S_m 裁剪后生成的综合语义缓存项记为 $S'_m=<S'_{Rm}, S'_{Am}, S'_{Pm}, S'_{Cm}>$。其中，$S'_{Rm}=S_{Rm}$，$S'_{Am}=\left(\bigcup_{i=1}^{n} S_{Ai}\right) \bigcap (Q_A \cup Q_{PA} \cup Q_{RK})$，$S'_{Pm}=S_{Pm}=\vee_{i=1}^{n} S_{Pi}$，$S'_{Cm}$ 是 S'_m 的实际内容。S'_m 中的元组若有空缺属性值，暂用 NULL 字符填入。

例 1　Q=“SELECT X1 FROM R1 WHERE ($X1$>10) and ($X3$>20)”，与该查询相关的语义缓存索引项有

$$S_1=<R_1, \{X_K, X_1\}, X_1>20, S_{C1}>$$

$$S_2 =< R_1,\{X_K,X_1,X_2,X_3\},X_3>30,S_{C2}>$$

其中，X_K 为 R_1 的主键。合并操作得到的语义缓存项 $S_m =< R_1,\{X_K,X_1,X_2,X_3\},(X_1>20)\vee(X_3>30),S_{Cm}>$，裁剪操作完成后生成的综合相关语义缓存项为 $S'_m =< R_1,\{X_K,X_1,X_3\},(X_1>20)\vee(X_3>30),S'_{Cm}>$。

综上所述，当移动终端与网络断开时，通过执行合并与裁剪操作生成的综合相关语义缓存项可以为当前查询返回近似结果。这一结果的正确性和完备性分析以及近似查询算法处理过程及计算复杂度分析详见第 3.3 节所述。

3.3 断接下的近似查询

3.3.1 正确性分析

定理 1 如果当前缓存数据有效，则综合相关语义缓存项导出的近似结果是精确解的一部分，即查询结果满足正确性。

证明 令当前断接查询 $Q =< Q_R,Q_A,Q_P,Q_C>$，合并和裁剪操作后生成的综合相关语义缓存项为 $S'_m =< S'_{Rm},S'_{Am},S'_{Pm},S'_{Cm}>$。服务器中执行 Q 得到的精确解为 $A=\{a_1,a_2,\cdots,a_n\}$，由 S'_m 导出 Q 的查询结果为 $X=\{x_1,x_2,\cdots,x_m\}$。

由于缓存数据有效，故有 $Q_C=A=\{a_1,a_2,\cdots,a_n\}$。

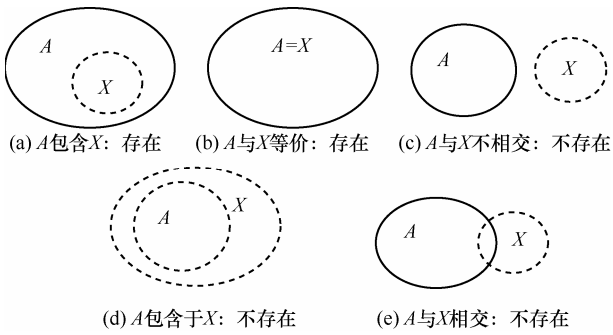
由定义 8 知，$Q_P\wedge S'_{Pm}\neq\Phi$ 且 $(Q_A\cup Q_{PA})\cap S'_{Am}\neq\Phi$，故 $\exists x$ 满足 $x\in S'_{Pm}$ 且 $x\in Q_C$，全部 x 组成的集合记为 X，则 $X\subseteq S'_{Cm}$ 且 $X\subseteq Q_C$。

由于 $Q_C=A$，故有 $X\subseteq A$。可见综合相关语义缓存项导出的结果是精确解的一部分，正确性满足。

证毕

由定理 1 可以推出，如果 S'_m 可以生成，则由 S'_m 执行查询的近似结果至少是正确不完备的，不存在不相交匹配的情况。图 1 给出了由 S'_m 导出的结果与精确解的各种相交情况，并分析如下。

图 1(a)和图 1(b)所表现的情况都是存在的，详见 3.3.2 节：其中图 1(a)表示断接下近似查询结果是精确结果的子集，即查询满足正确性但不满足完备性；图 1(b)表示断接下近似查询结果就是精确结果，查询同时满足正确性和完备性。图 1(c)~图 1(e)所表现的情况都是不存在的：其中图 1(c)表示断接下近似查询结果与精确解没有交集，查询既不满足正确性，也不满足完备性；图 1(d)表示断接下近似查询结果包含精确解，查询满足完备性，但不满足正确性；图 1(e)表示断接下近似查询结果与精确解相交，查询既不满足正确性，也不满足完备性。



(a) A包含X：存在 (b) A与X等价：存在 (c) A与X不相交：不存在

(d) A包含于X：不存在 (e) A与X相交：不存在

图 1 S'_m 的导出结果 X 与精确解 A 的关系

3.3.2 完备性分析

由定理 1 可知，终端断接近似查询结果是正确的，但有可能不完备，本节从以下两方面来讨论完备性。

1) 近似查询结果既正确又完备

当离线近似查询结果与在线精确结果一致(或等价)时，称其为正确且完备的查询。这是最理想的情况，不过通常发生的概率较小，即有些文献[9]中所说的精确匹配。

若 Q 和 S'_m 满足条件 $(Q_A\cup Q_{PA}\cup Q_{RK})=S'_{Am}$ 和 $Q_P=S'_{Pm}$，则说由 S'_m 返回的近似结果 X 与精确结果 A 等价，此时 $X=A=\pi_{Q_A}\sigma_{Q_P}(S'_m)$。等价查询结果近似度等于 1。

2) 近似查询结果正确但不完备

当离线近似查询结果是精确解的一部分时，称其为正确但不完备的查询，这是最常见的一种情况，即有些文献[9]中所说的包含匹配和相交匹配。

若 Q 和 S'_m 满足条件 $(Q_A\cup Q_{PA})\cap S'_{Am}\neq\Phi$ 和 $Q_P\wedge S'_{Pm}\neq\Phi$，则说由 S'_m 返回的近似结果 X 是精确结果 A 的一部分，此时 $X=\pi_{(Q_A\cap S'_{Am})}\sigma_{Q_P}(S'_m)\subseteq A$。近似度介于 0 和 1 之间，计算方法见定义 10。

3.3.3 算法过程及复杂性分析

SQPID 算法流程如图 2 所示，伪码如图 3 所示。SQPID 算法基本思想是先判断出所有的相关语义缓存项，这一过程的时间复杂度为 $O(n)$，其中 n 为终端语义缓存索引项数目；然后执行合并和裁剪操作生成综合相关语义缓存项 S'_m，这一过程的时间复杂度为 $O(m)$，m 为相关语义缓存索引项数目；最后在 S'_m 上执行查询得到近似结果，这一过程的时间复杂度为 $O(1)$。综上可见，算法总体时间复杂度为 $O(n+m+1)$，是线性多项式级别，属 P 类问题。

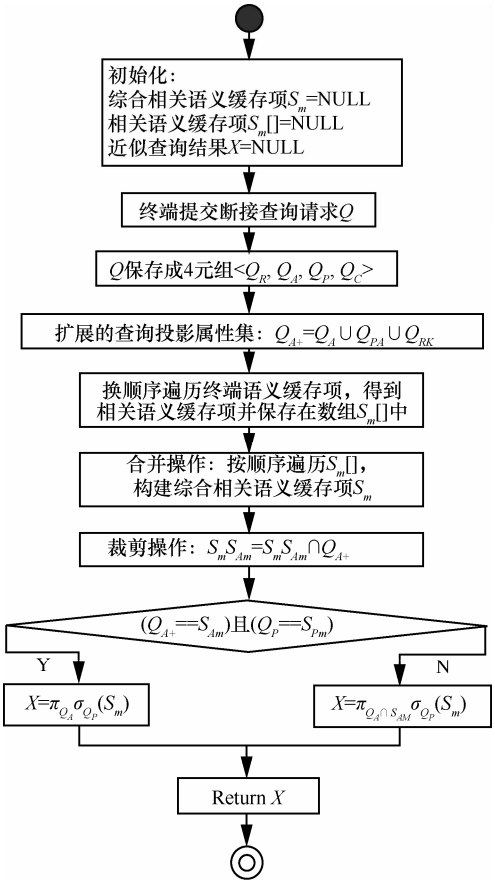


图 2　SQPID 算法流程

Name：SQPID(Query Q，SemanCache S，QueryResult X)
Input：S，Q
Output：X
//***S：semantic cache of mobile client***//
//***Q：query request in disconnection***//
//***X：Q's approximate result***//
Procedure：
{
S_m←NULL；
S_m[]←NULL；
X←NULL；
Q←<Q_R,Q_A,Q_P,Q_C>；
Q_{A+}←$Q_A \cup Q_{PA} \cup Q_{RK}$；
FOR(S_i：S){ //***find Q's related semantic cache items in S***//
IF(($S_{iR}=Q_R$) and (($Q_A \cup Q_{PA}$)$\cap S_{iA} \neq \Phi$) and ($Q_P \wedge S_{iP} \neq \Phi$))
$S_m[i]=S_i$；}
FOR(S_i：S_m[]){//***combine operation***//
$S_m.S_{Am}=S_m.S_{Am} \cup S_i.S_{Ai}$；
$S_m.S_{Pm}=S_m.S_{Pm} \vee S_i.S_{Pi}$；}
$S_m.S_{Am}=S_m.S_{Am} \cap Q_{A+}$；//***clip operation***//
S_m=<S_{Rm},S_{Am},S_{Pm},S_{Cm}>；
IF (($Q_{A+}=S_{Am}$)and($Q_P=S_{Pm}$))
$X=\pi_{Q_A}\sigma_{Q_P}(S_m)$；
ELSE $X=\pi_{(Q_A \cap S_{Am})}\sigma_{Q_P}(S_m)$；
RETURN X；
}

图 3　断接下简单查询算法 SQPID 的伪码

4　实验及数据分析

4.1　实验平台及参数设置

为了对算法性能进行验证及性能分析，编程实现了文中算法和查询模拟器，并设计了几组实验进行性能测试。实验中所用的主要参数及取值如表 2 所示。编程工具是 JDK 1.7 开发分组，运行环境是双核 Pentium 2.00 GHz 处理器，2 GB 内存，Windows XP Professional 操作系统。

表 2　　实验参数设置

Parameter	Value and Description
*R*1	关系表 *R*1：包含 1 000 条元组，6 个属性列，其中第 1 列为主键
*R*2	关系表 *R*2：包含 1 200 条元组，8 个属性列，其中第 1 列为主键
*R*3	关系表 *R*3：包含 1 500 条元组，12 个属性列，其中第 1 列为主键
DisconQuery	50~450:断接下终端提交的查询请求数
ConnQuery	450~50:在线时终端提交的查询请求数
SumQuery	500：终端查询请求总数
approx	[0，1]：断接下查询的近似度(精确度)
TimeADQ	断接下查询响应时间均值
TimeDQ	断接下查询响应时间

实验平台中包括的模块有：查询产生器、数据库生成器、算法模拟器和语义缓存生成器。各模块关系如图 4 所示。

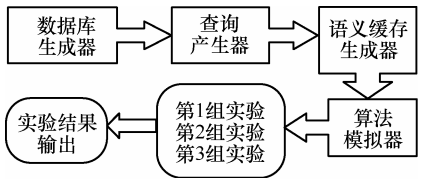


图 4　实验平台中各模块关系

1) 数据库生成器

数据库生成器负责产生数据库 D 中关系表{*R*1，*R*2，*R*3}的内容，并为各关系表中的属性分配选择概率，以及对数据热区范围进行定义。

2) 查询产生器

查询产生器负责根据热点属性分布、投影属性选择概率、元组数据热区和查询语义相关性等参数，在数据库 D 中随机产生 *SumQuery*=500 个查询。

3) 语义缓存生成器

语义缓存生成器的任务是执行查询产生器随机生成的 SumQuery 个查询，并利用前 ConnQuery

个在线查询结果构建语义缓存数据库。由于缓存空间有限，创建缓存过程中还涉及到缓存项合并及替换操作。

4) 算法模拟器

算法模拟器负责根据不同的实验场景，设置相应的实验参数，对 SQPID 算法性能进行分析，例如综合相关语义缓存项构建时间、查询响应时间和近似度等。

4.2 实验结果及分析

4.2.1 实验 1：时间性能分析

算法模拟程序执行总数为 *SumQuery*=500 个查询，其中断接查询数与在线查询数比率从1:9 逐渐变至 9:1，可见用于构建缓存数据库的在线查询数逐渐减少。图 5 和图 6 分别描绘了 SQPID 算法和 QPID 算法在断接近似查询过程的平均耗时和综合相关语义缓存项构建耗时两方面的比较情况。

如图 5 所示，随着在线查询数的递减，2 种算法的查询响应时间均呈减少趋势，但 SQPID 算法的响应时间更短。对图中 9 种情况下响应时间求取平均值后，发现 SQPID 算法的查询时间均值为 0.562 5 s，而 QPID 算法的响应时间均值为 2.526 s。这是由于 QPID 算法的语义缓存项中仅对投影属性集和关系表主键进行保存。所以在寻找相关语义缓存项时，需要同时判断直接相关项、间接相关项和进行等价性判断，从而大大增加了时延。相对而言，SQPID 算法的语义缓存项对关系表主键、条件属性集和投影属性集都进行保存。这样虽然会带来一定的空间开销，但算法更容易实现，且能简化综合相关语义缓存项的生成过程，减少断接近似查询响应时间。

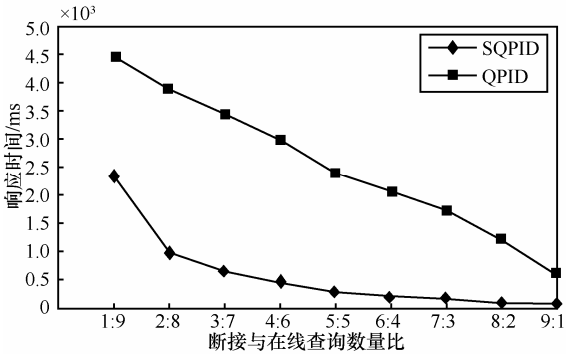


图 5 2 种算法的断接查询响应时间比较

如图 6 所示，随着断接查询数的增加，综合相关语义缓存项构建时间在减少，原因是在线查询数的减少，使本地缓存项数目也在减少。另外，SQPID 算法较 QPID 算法有更低的时间开销，这是由于 SQPID 在构建综合相关语义缓存项时无需进行间接相关语义缓存索引的判断、转换和合并过程，从而加快了查询结果的导出速度。实验数据表明，SQPID 算法处理每条语义缓存索引项平均需 0.294 s，约占断接查询处理总时间的 52%。

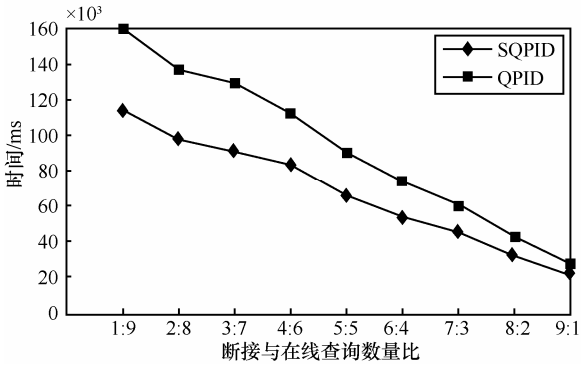


图 6 2 种算法的综合相关语义缓存项构建耗时比较

4.2.2 实验 2：近似度性能分析

如图 7 所示，随着断接查询数量的增加，查询近似度逐渐降低。当断接查询与在线查询数量之比达到 7:3 时，近似度下降幅度增大，当达到 9:1 时达到最低值 0.947。不过这一数据并不绝对，与当前数据库中关系表数以及查询语句相关性有较大关系。例如当比率为 9:1 时，其中有 50 个在线查询，这一数量远大于数据库关系表数和表中平均属性数量。查询语句的语义相关性较大，能够构建较为完整的语义缓存数据库，因此断接查询的近似度较高。

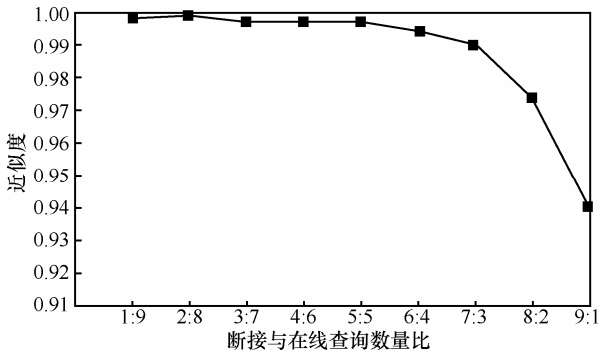


图 7 断接查询的近似度变化

4.2.3 实验 3：等价查询比例性能分析

如图 8 所示，随着断接近似查询数量的递增，断接近似查询结果与在线查询结果等价的数量也在递减。这是因为在线查询越少，由其构建的终端语义缓存数据库较服务器中的原始数据库就越不

完整，从而影响等价查询率。在断接查询与在线查询数量之比为 6:4 时，之后等价查询率减幅变大，在 9:1 时降低到 76%，算法过程的平均等价查询率为 90%。同样值得说明的是，这一数据并不绝对，与当前数据库中关系表数以及查询语句相关性有较大关系，查询语句间的语义相关性越高，断接下等价查询率越高，近似查询的效果越好。

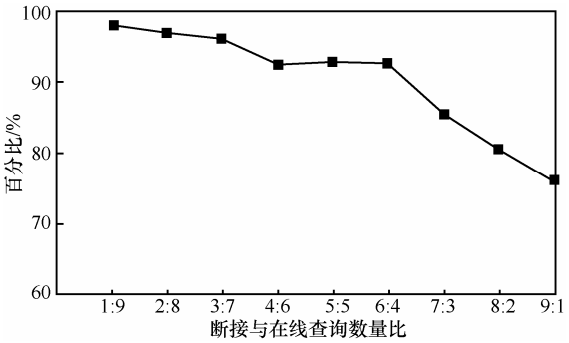


图 8　等价查询率变化

由模拟实验 2 和实验 3 得到的数据进一步分析可得出断接下查询的完备性分析柱形图，如图 9 所示。随着断接查询数的逐渐增加，不完备查询数也在增加，等价查询数减少，利用终端缓存完全得不到结果的查询数最多占 6%，还属于用户可接受的范围。可见，利用语义缓存进行断接下近似查询能够满足用户的需求。

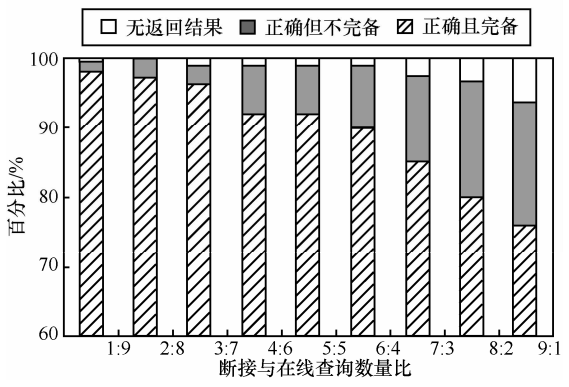


图 9　断接查询的完备性分析柱形图

5　结束语

移动环境的弱连接和资源有限性，使时刻为移动终端提供精确查询很难，这需要付出能量和流量的双重代价。而另一方面，在用户可接受的时间、精度内，提供的近似查询结果也可以满足部分用户，从而增加了系统的可用性。利用终端的语义缓存可以在断接或弱连接时提供查询的近似结果，是实现这一目标的重要方法，但目前尚存在查询处理时间长和缓存项相关性判断过程复杂的问题。

本文提出断接下移动终端简单查询算法 SQPID，对算法中涉及的相关定义、算法过程及复杂度进行了详细的阐述和分析。该算法基本思想是通过合并与裁剪操作构建综合相关语义缓存项，合并过程不涉及间接相关性判断，从而简化了以往算法的处理过程，加快了近似查询结果的导出速度。实验数据表明，SQPID 算法在查询响应时间和精确度方面都更好地满足了用户的需求。进一步研究工作包括弱连接下的近似查询算法，以及断接下位置相关查询算法。

参考文献：

[1] LIANG R B, LIU Q. A technology of agent-cache used in mobile computing environment[A]. World Congress on Computer Science and Information Engineering[C]. 2009.120-124.

[2] 梁茹冰,刘琼.使用 MSS 维护语义缓存一致性的方法[J]. 华南理工大学学报(自然科学版), 2011, 39(7): 127-133.
LIANG R B, LIU Q. Maintenance of semantic cache coherency via mobile support station[J]. Journal of South China University of Technology(Natural Science Edition), 2011, 39(7): 127-133.

[3] DAR S, FRANKLIN M, JONSSON B, *et al*. Semantic data caching and replacement[A]. Proceedings of the 22nd VLDB Conference. Mumbai[C]. 1996.330-341.

[4] REN Q, DUNHAM M H, KUMAR V. Semantic caching and query processing[J]. IEEE Transactions on Knowledge and Data Engineering, 2003, 15(1): 192-210.

[5] REN Q, DUNHAM M H. Using semantic caching to manage location dependent data in mobile computing[A]. Proceedings of the 6th Annual International Conference on Mobile Computing and Networking[C]. New York, 2000.210-221.

[6] 吴婷婷, 周兴铭. 基于语义缓存的移动查询导出[J]. 计算机学报, 2002, 25(10): 1104-1110.
WU T T, ZHOU X M. Extracting query results from semantic cache[J]. Chinese Journal of Computers, 2002, 25(10): 1104-1110.

[7] 吴婷婷, 苏武运, 周兴铭等. 移动查询缓存处理的研究[J]. 计算机研究与发展, 2004, 41(1): 187-193.
WU T T, SU W Y, ZHOU X M, *et al*. Mobile query through semantic cache[J]. Journal of Computer Research and Development, 2004, 41(1): 187-193.

[8] 蔡建宇, 吴泉源, 贾焰等. 面向聚集查询的语义缓存技术[J]. 软件学报, 2007, 18(2): 361-371.
CAI J Y, WU Q Y, JIA Y, *et al*. Semantic cache technology for aggregate queries[J]. Journal of Software, 2007, 18(2): 361-371.

（下转第 215 页）

ropic diffusion[J]. IEEE Trans Pattern Analysis and Machine Intelligence, 1990, 2(7):629-639.

[15] YOU Y, TANNENBAUM A, KAVEH M. Behavioral analysis of anisotropic diffusion in image processing[J]. IEEE Trans Image Processing, 1996, 5(11):1539-1553.

[16] DO M, VETTERLI M. Framing pyramids[J]. IEEE Trans Signal Processing, 2003, 51(9):2329-2342.

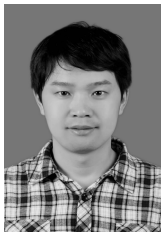
[17] RARCE G, TIAN M. Order statistic filter banks[J]. IEEE Trans Image Processing, 1996, 5(6):827-837.

[18] http://www.cbsr.ia.ac.cn/irisdatabase.htm.

[19] http://pesona.mmu.edu.my/~ccteo/.

[20] http://iris.di.ubi.pt/.

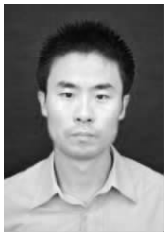
作者简介：



万洪林（1979-），男，山东德州人，博士，山东师范大学讲师，主要研究方向为图像处理、生物特征识别。

李宝生（1965-），男，山东潍坊人，山东省肿瘤医院教授、博士生导师，主要研究方向为肿瘤精确放射治疗和综合治疗。

韩民（1973-），男，山东菏泽人，博士，山东大学副教授，主要研究方向为信号与图像处理、生物特征识别。



李登旺（1983-），男，山西吕梁人，博士，山东师范大学副教授，主要研究方向为生物医学工程。

（上接第 207 页）

[9] 吴婷婷, 章文嵩, 周兴铭. 断接下查询的缓存处理[J]. 计算机学报, 2003, 26(10): 1393-1399.
WU T T, ZHANG W S, ZHOU X M. Answering query through cache during disconnection[J]. Chinese Journal of Computers, 2003, 26(10): 1393-1399.

[10] 陈良刚, 孙未未, 施伯乐. 位置相关查询处理的近似回答[J]. 计算机研究与发展, 2003, 40(11): 1593-1597.
CHEN L G, SUN W W, SHI B L. Approximate answers in location dependent query processing[J]. Journal of Computer Research and Development, 2003, 40(11): 1593-1597.

[11] 何文麟, 陈红. 传感器网络中多近似连续范围查询的处理技术[J]. 计算机研究与发展, 2010, 47(5): 754-761.
HE W L, CHEN H. Multi-query processing technology of approximate continuous queries in wireless sensor networks[J]. Journal of Computer Research and Development, 2010, 47(5): 754-761.

作者简介：

梁茹冰（1980-），女，安徽肥东人，博士，华南农业大学讲师，主要研究方向为移动计算、语义缓存。

刘琼[通信作者]（1959-），女，云南昆明人，博士，华南理工大学教授、博士生导师，主要研究方向为计算机网络、移动计算、模式识别。E-mail: liuqiong@scut.elu.cn。