有效降低分布式 SKYLINE 查询网络传输代价

黄震华[1,2],向 阳[1],林 琛[3]
(1.同济大学计算机科学与工程系,上海 200092; 2.同济大学嵌入式系统与服务计算教育部重点实验室,上海 200092;
3.复旦大学计算机与信息技术系,上海 200433)

摘 要: 分布式网络中的 skyline 查询是近年来信息检索学科的一个研究重点.目前大多数研究工作均没有考虑在分布式网络中,如何有效降低 skyline 查询的网络传输代价.为此,提出一种在分布式网络中,有效降低 skyline 查询传输代价的方法 RTCSQDN(Reducing the Transferring Cost of Skyline Queries over Distributed Networks).RTCSQDN 算法充分利用父空间 skyline 对象集与子空间 skyline 对象集间的语义关系通过三个阶段来平衡网络传输量和查询时间开销.同时,文章给出一种新颖的多维对象传送策略 PTGPV(Policy for Transferring Grouping Position Values)来避免直接传送 skyline 对象本身,从而最小化数据传输量.详细的理论分析和大量实验评估表明,文章给出的算法具有有效性和实用性.

关键词: 信息检索; skyline 查询; 分布式网络; 传输代价
中图分类号: TP311.13 **文献标识码:** A **文章编号:** 0372-2112 (2010) 04-0848-05

Efficiently Reducing Transferring Cost of Skyline Queries over Distributed Networks

HANG Zhen-hua[1,2], XIANG Yang[1], LIN Chen[3]
(1. *Department of Computer and Technology, Tongji University, Shanghai* 200092, *China*;
2. *The Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji University, Shanghai* 200092, *China*;
3. *Department of Computer and Information Technology, Fudan University, Shanghai* 200433, *China*)

Abstract: Skyline query processing in distributed environments has recently received a lot of attention in information retrieval community. However, most existing literatures do not consider how to efficiently reduce the transferring cost of skyline quereis on distributed networks. In this paper, we propose RTCSQDN(Reducing the Transferring Cost of Skyline Queries over Distributed Networks), the efficient sound and complete algorithm for balancing the cost of transferring data and skyline computation cost on the distributed networks. Specially, the PDSQDN algorithm makes use of the semantic relationship between parent-space skylines and child-space skylines, and transfers the data through three phases. Moreover, we present a novel policy, i.e. PTGPV (Policy for Transferring Grouping Position Values), to transfer multi-dimensional objects. The PTGPV policy only transfers the position values of objects for most dimensional-spaces, and hence it can efficiently minimize the volume of data transferred. We also present detailed theoretical analyses and extensive experiments that demonstrate our algorithms are both efficient and effective.

Key words: information retrieval; skyline query; distributed networks; transferring cost

1 引言

Skyline 查询处理技术是近年来信息检索领域的一个研究重点[2~5,7,8,11,14,15].给定对象集合($\psi=\{p_1,\cdots,p_n\}$,其中每个对象 p_i 具有 w 个维度,skyline 查询就是在(中获取一类对象集合 SKY,它满足如下条件:$SKY\subseteq\psi$ 且 SKY 中每个对象不会在所有维度上的取值均差于(中的某一对象.值得注意的是,根据用户的查询偏好,"差于"可以有两种选择,即"<"或">".

随着分布式网络系统的普及和深入应用,文献[1,6,9,10]基于分布式网络[12]给出有效处理 skyline 查询的分布式算法(也叫分布式 skyline 查询算法).在分布式网络架构中,LP 终端与 HP 服务器间的数据传输开销,构成分布式 skyline 查询的网络传输代价.文献[6,9,10]在数据传输时,简单地将所有存储在 LP 终端上的数据传输到 HP 服务器上.显然,在实际应用中,由于非主干网的带宽比较有限,因此,传输所有数据的方法是不现实的.为此,文献[1]给出"Eskyline 集合"来避免传输

收稿日期:2009-03-03;修回日期:2009-05-25
基金项目:国家自然科学基金(No.60903032,70771077);教育部博士点基金(No.20090072120056);国家 863 高技术研究发展计划(No.2008AA04Z106);上海市科委科研计划项目(No.08DZ1122300);同济大学青年优秀人才基金(No.0800219093)

所有的数据.然而,当分布式 skyline 查询个数较少时,Eskyline 集合中的大多数对象均是多余的,即它们不可能出现在任一用户查询的返回结果中;因此完全没必要将这些对象从 LP 终端传输到 HP 服务器上.另外,Eskyline 集合没有对数据对象进行编码压缩,即 LP 终端与 HP 服务器间通过传输多维对象本身来完成..因此,当对象具有多个维度时,Eskyline 集合的缩减能力变得极其低下.

为了有效解决分布式 skyline 查询的网络传输性能问题,本文提出了 RTCSQDN(Reducing the Transferring Cost of Skyline Queries over Distributed Networks)算法.RTCSQDN 算法充分考虑到用户提交的分布式 skyline 查询的内容和个数,只传输与查询相关的数据对象.这样,对于分布式 skyline 查询,LP 终端无需将大量多余的对象传输到 HP 服务器,从而在很大程度上降低了网络传输量.而且,对于需要传输的数据对象,RTCSQDN 算法避免传送多维对象本身,而是通过将多维对象实体映射为一维的位置值信息来最小化分布式 skyline 查询的网络传输量.此外,RTCSQDN 算法充分考虑到 LP 终端和 HP 服务器间计算能力的差异,将网络传输过程划分为三个阶段,特别,算法将计算量较小的第 1 阶段和第 3 阶段放在 LP 终端上执行,而将 CPU 敏感的第 2 阶段放在高性能的 HP 服务器上完成.本文的理论分析和实验评估表明,RTCSQDN 算法具有实用性和有效性.

2　概念和术语

定义 1(维空间)　假定对象集合 AD 总共具有 k 个考察维度 $d_1,\cdots,d_k$,那么每种考察维度的组合称为一个维空间.并且,我们把所有包含 $v(0<v\leqslant k)$个维度的维空间统称为 v 维空间.

不难看出,一个包含 k 个考察维度的对象集合,它最多具有 2^k-1 个维空间($\varnothing$ 除外).此外,我们把含有所有 k 个维度的维空间称为全空间,记为 F.

定义 2(支配关系)　假定 p 和 r 是两个具有 k 个维度的对象,如果它们满足下列两个条件,那么我们称 p 在维空间 $V(V\subseteq F)$上支配 r:① $\forall\alpha\in V, p[\alpha]\leqslant r[\alpha]$;② $\exists\beta\in V, p[\beta]<r[\beta]$.

为了简单,我们把 p 在 V 上支配 r,记为 $r\prec_V p$.

定义 3(skyline 集合)　设 AD 是 k 维对象全集,那么 $v\ (0<v\leqslant k)$维空间 V 上的 skyline 集合 $\nabla^V(AD)$可表示为:$\nabla^V(AD)=\{p\mid p\in AD\wedge\neg\exists\, r\in AD, p\prec_V r\}$.

定义 4(扩展支配关系)　假定 p 和 r 是两个具有 k 个维度的对象,如果它们满足如下条件,则我们称 p 在维空间 $V(V\subseteq F)$上扩展支配 r:$\forall\alpha\in V, p[\alpha]<r[\alpha]$.

不难看出,如果 p 扩展支配 r,那么 p 必定支配 r;反之不成立.

定义 5(Eskyline 集合)　设 AD 是 k 维对象全集,那么 $v(0<v\leqslant k)$维空间 V 上的 Eskyline 集合 $\Delta^V(AD)$可表示为:$\Delta^V(AD)=\{p\mid p\in AD\wedge\neg\exists\, r\in AD, p\prec_V r\}$.

根据定义 3 和 5 可知,维空间 V 上的 skyline 集合是 Eskyline 集合的子集,即 $\nabla^V(AD)\in\Delta^V$.

定义 6(种子 skyline 集合)　假定维空间 V 的父空间为 U(即 $V\subset U$),并且 $\nabla^U(AD)$为数据集 AD 中,在维空间 U 上的 skyline 对象集合,那么 V 上的(AD,U)-种子 skyline 集合 $seed(AD,U,V)=\nabla^V(\nabla^U(AD))$.

在不引起混淆的情况下,为了简单,我们省略去(AD,U)这两个参数,把 $seed(AD,U,V)$写成 $seed(V)$.

3　RTCSQDN 算法

这一节提出一种控制分布式 skyline 查询网络传输代价的有效方法 RTCSQDN.算法 RTCSQDN 将网络数据传输过程划分为三个阶段.不失一般性,我们假定 LP 终端上的对象集为 AD,用户发出 w 个维空间 $V_1,\cdots,V_w$ 上的 skyline 查询分别为 $SQ(V_1),\cdots,SQ(V_w)$.

3.1　RTCSQDN 算法第 1 阶段

在 RTCSQDN 算法的第 1 阶段,我们基于现有的 skyline 查询计算方法 SFS[3]在 LP 终端上获取维空间 $US=\bigcup_{i=1}^{w}V_i$ 上的 skyline 集合 S_g,并将 S_g 传送给 HP 服务器.第 1 阶段算法描述如下.

算法 1:RTCSQDN-Ⅰ

输入:维空间 $US=\bigcup_{i=1}^{w}V_i$;对象集合 AD.
输出:维空间 US 上的 skyline 集合 S_g.
beign
1.　$S_g\leftarrow\varnothing$;
2.　对 AD 按维度 $\alpha\in US$ 非递减排序;
3.　for 顺序访问 AD 中的每个对象 p do
4.　　if $\neg\exists\, r\in S_g, p\prec_{US} r$ then
5.　　$S_g\leftarrow S_g\cup\{p\}$;
6.　return S_g;
end

接下来,我们具体分析 RTCSQDN 算法第 1 阶段的数据传输量和 CPU 时间代价.令 $\lambda=|US|$.

定理 1　设 AD 在 $US=\bigcup_{i=1}^{w}V_i$ 上满足联合分布函数 $F(\bar{x})$和联合密度函数 $f(\bar{x})$,其中 $\bar{x}=(x_1,\cdots,x_\lambda)$,那么 S_g 中对象数的期望值 $E(|AD|,\lambda)$可表示为[16]:

$$|AD|\times\int_{[0,1]^\lambda}f(\bar{x})(1-F(\bar{x}))^{|AD|-1}\mathrm{d}\bar{x}.$$

假定每个维度占 ξ 个字节,根据定理 1 可知第 1 阶段的数据传输量为 $\xi\times\lambda\times E(|AD|,\lambda)$字节.

根据文献[16]的评估思想,我们可以得到 RTCSQDN 算法第 1 阶段所需对象间比较的次数,如定理 2 所示.

定理 2 设 AD 在 $US=\bigcup_{i=1}^{w}V_i$ 上满足联合分布函数 $F(\bar{x})$和联合密度函数 $f(\bar{x})$,其中 $\bar{x}=(x_1,\cdots,x_\lambda)$,那么 RTCSQDN 算法第 1 阶段所需对象间比较的次数 $C_{\mathrm{I}}(|AD|,\lambda)$可表示为

$$\sum_{j=2}^{|AD|}\frac{E(j-1,\lambda)\times E(j-1,\lambda+1)}{j-1}.$$

3.2 RTCSQDN 算法第 2 阶段

当 HP 服务器接收到 LP 终端传送过来的对象集合 S_g 之后,算法首先将 w 个查询组织成序列,然后基于 S_g,沿着每条树路径,利用各节点所对应的 skyline 集合间的语义关系来获取这 w 个分布式 skyline 查询相应的种子 skyline 对象集合 $seed(V_1),\cdots,seed(V_w)$,并将这 w 个种子 skyline 对象集合传送给 LP 终端.RTCSQDN 算法第 2 阶段描述如下:

算法 2:RTCSQDN-Ⅱ

输入:w 个维空间 $V_1,\cdots,V_w$;对象集合 S_g.
输出:w 个位置值列表 $list(V_1),\cdots,list(V_w)$.
beign
1. $DTS\leftarrow w$ 个维空间组织成的维空间树序列;
2. for DTS 中的每个维空间树 T_{sb} do
3. for 广度优先访问 T_{sb}中的每个节点 nd do
4. if nd 是根节点 then
5. $seed(nd)\leftarrow SFS(nd,S_g)$;
6. else
7. $pd\leftarrow nd$ 的父节点;
8. $seed(nd)\leftarrow SFS(nd,seed(pd))$;
9. $seed(pd)\leftarrow seed(pd)\text{-}seed(nd)$;
10. for $i=1$ to w do $list(V_i)\leftarrow\emptyset$;
11. for 每个维空间 V_i do
12. for $seed(V_i)$中的每个对象 r do
13. $pos(r)\leftarrow r$ 在 S_g 中的位置值;
14. $list(V_i)\leftarrow list(V_i)\cup\{pos(r)\}$;
15. 按 $pos(r)$非递减排序 $list(V_i)$中的位置值;
16. 返回 w 个位置值列表 $list(V_1),\cdots,list(V_w)$.
end

接下来,我们具体分析 RTCSQDN 算法第 2 阶段的数据传输量和 CPU 时间代价.

引理 1 假定 rt 为维空间树 T_{sb}的根节点.则对于 T_{sb}上的每个节点 nd,它的种子 skyline 对象集合的基数的期望值 $EC(nd)$可用下列的递归式来表示(pd 为 nd 的父节点,$|nd|$为 nd 的维数):

$$EC(nd)=\begin{cases}E(|S_g|,|nd|) & ,\text{iff } nd=rt\\ E(EC(pd),|nd|) & ,\text{otherwise}\end{cases}$$

定理 3 假定维空间树序列 DTS 包含 u 棵维空间树 $T_{sb}^{(0)},\cdots,T_{sb}^{(u)}$.对于 $T_{sb}^{(i)}(1\leqslant i\leqslant u)$,假定它具有 θ_i 个叶子节点 $lf_1,lf_2,\cdots,lf_{\theta_i}$.那么 RTCSQDN 算法第 2 阶段传送的位置值个数 $PN(DTS)$满足如下不等式:

$$\begin{cases}PN(DTS)\leqslant\sum_{T_{sb}\in DTS}\sum_{nd\in T_{sb}}EC(nd)\\ PN(DTS)\geqslant\sum_{i=1}^{u}\sum_{j=1}^{\theta_i}EC(lf_j)\end{cases}$$

定理 4 假定维空间树序列 DTS 包含 u 棵维空间树 $T_{sb}^{(0)},\cdots,T_{sb}^{(u)}$.对于 $T_{sb}^{(i)}$,假定它的节点数为 $\lambda(i)$,根节点为 $rt(i)$.那么 RTCSQDN 算法第 2 阶段所需对象间比较的次数 $C_{\mathrm{II}}(DTS)$为:

$$\sum_{i=1}^{u}\{C_{\mathrm{I}}(|S_g|,|rt(i)|)+\sum_{j=2}^{\lambda(i)}C_{\mathrm{I}}(EC(pd_j),|nd_j|)$$

3.3 RTCSQDN 算法第 3 阶段

当 LP 终端接收到 HP 服务器传送过来的 w 个列表之后,对于每个 $seed(V_i)$,RTCSQDN 算法在 $AD\text{-}seed(PV_i)$对象集中获取与 $seed(V_i)$在 V_i 上重复的对象集 $rep(V_i)$,其中 PV_i 为 V_i 的父节点.为了提高 RTCSQDN 算法在该阶段的效率,我们将 $AD\text{-}seed(PV_i)$集合划分为两部分:① $S_g\text{-}seed(PV_i)$集合,以及② $AD\text{-}S_g$ 集合.对于①,由于 $S_g\text{-}seed(PV_i)$远小于 $AD\text{-}S_g$,因此对于这部分对象集,我们直接通过简单的维度值比较来获取重复对象即可.对于②,由于 $AD\text{-}S_g$ 中包含大量的对象,以及它对于各个种子 skyline 集合均是一样的,因此我们为 $AD\text{-}S_g$ 集合在$\bigcup_{i=1}^{w}V_i$ 的每个维上建立 B-树索引,这样对于多次查询来说,可以具有索引重用的优点[13].

值得注意的是,w 个维空间所包含的维度在很大程度上存在重复性,因此,RTCSQDN 算法采用对象和维度共享策略来提高查找重复对象的效率,具体做法可描述如下.我们创建一个二维网格的数据结构 TM. TM 包含$|\bigcup_{i=1}^{w}seed(V_i)|$个行和$|\bigcup_{i=1}^{w}V_i|$个列,每一行代表$\bigcup_{i=1}^{w}seed(V_i)$中的一个对象 r,每一列表示$\bigcup_{i=1}^{w}V_i$ 中的一个维度 α,而单元格 $TM(r,\alpha)$包含在 $AD\text{-}S_g$ 集合中所有与 r 在维度 α 上取值均相同的对象组成的集合.RTCSQDN 算法初始化 TM 的各单元格均为 $\emptyset$.然后,对 $seed(V_i)(1\leqslant i\leqslant w)$中的每个对象 r 以及 V_i 中的每个维度 α,基于 B-树索引获取 r 在 α 上的重复对象集合.如果 $TM(r,\alpha)$不为空,那么不需要进行处理,否则需要获取相应的重复对象集合,并填充单元格 $TM(r,\alpha)$.我们不难看出,$\bigcap_{\alpha\in V_i}TM(r,\alpha)$为 r 在维空间 V_i 上的重复对象集合.从而,$seed(V_i)$在 $AD\text{-}S_g$ 中的重复对象集合 $rep(V_i)=\{\bigcap_{\alpha\in V_i}TM(r,\alpha)\mid r(seed(V_i))\}$.

与 PTGPV 传送策略类似,为了最小化 RTCSQDN 算法在第 3 阶段的数据传输量,我们不传送多维对象本身,而只传送对象在 S_g 中的位置值信息.另一方面,如果$\bigcap_{\alpha\in V_i}TM(r,\alpha)$中包含 t 个重复对象,我们无需将这 t 个重复对象的位置值都传送给 HP 服务器,只需传送格式为$\langle pos(r),t\rangle$的信息项即可.

根据以上的分析,我们给出完成 RTCSQDN 算法第

3 阶段工作的伪码,如算法 3 所示.

接下来,我们具体分析 RTCSQDN 算法第 3 阶段的数据传输量和 CPU 时间代价.

定理 5　假定 AD 的平均重复度为 δ,每个位置值和计时器分别占用 ξ 和 χ 字节,那么 RTCSQDN 算法第 3 阶段的数据传输量 $DV(DTS)$ 为:

$$\delta\times(\xi+\chi)\times\sum_{i=1}^{w}EC(V_i).$$

定理 6　假定维空间树序列 DTS 包含 u 棵维空间树 $T_{sb}^{(0)},\cdots,T_{sb}^{(u)}$. 对于 $T_{sb}^{(i)}$,假定它的根节点为 $rt(i)$. 那么 RTCSQDN 算法第 3 阶段所需对象间比较的次数 $C_{\text{Ⅲ}}(DTS)$ 为(令 V_i 的父节点为 PV_i):

$$\sum_{i=1}^{u}EC(rt^{(i)})\times|rt^{(i)}|\times\log|AD-S_g|+\sum_{i=1}^{w}EC(V_i)\times(|S_g|-EC(PV_i))$$

算法 3:RTCSQDN-Ⅲ

输入:对象集 S_g;w 个位置值列表 $list(V_1),\cdots,list(V_w)$.

输出:w 个重复对象位置值列表 $rep(V_1),\cdots,rep(V_w)$.

```
beign
1.  创建|∪_{i=1}^{w} seed(V_i)|行和|∪_{i=1}^{w} V_i|列二维网格 TM;
2.  初始化 TM 的各单元格均为 ∅;
3.  for_i = 1 to w do
4.    rep(V_i) ← ∅;
5.    基于 list(V_i)获取 seed(V_i);
6.    for seed(V_i)中的每个对象 r do
7.      count ← 0;
8.      pos(r) ← r 在 S_g 中的位置值;
9.      for S_g-seed(PV_i)中的每个对象 p do
10.       if ∀α∈V_i, r[α] = p[α] then count ++ ;
11.     for V_i 中的每个维度 α do
12.       if TM(r,α) = ∅ then
13.         TM(r,α) ← r 在 α 上的重复对象集;
14.   count(count + |∩_{α∈Vi} TM(r,α)|;
15.   rep(V_i) ← rep(V_i)∪{<pos(r), count>};
16. 返回 w 个列表 rep(V_1),…,rep(V_w).
end
```

3.4　RTCSQDN 算法的正确性和完备性

在这一小节中,我们给出 *RTCSQDN* 算法的正确性和完备性的证明,如定理 7 所示.

定理 7　假定 AD 是 k 维对象全集,V_t 是节点 V_j 在维空间树 T_{sb} 上的父节点. 同时,我们假设 $CA(V_t)$ 和 $CA(V_j)$ 分别为 RTCSQDN 算法在第 1 阶段产生的维空间 V_t 与 V_j 上的 skyline 对象集合. 那么算法结束时,维空间 V_j 上的 skyline 集合((V_j) 由且仅由如下两部分构成:(1) $CA(V_j)$;和(2) $AD\text{-}CA(V_t)$ 对象集中,与 $CA(V_j)$ 内的对象在 V_j 上取值均相同的对象组成的集合 $RP(V_j)$,即 $RP(V_j)=\{p\mid p(AD\text{-}CA(V_t))(r(CA(V_j)),(z(V_j,p[z]=r[z]\}$.

证明　假定根节点为第 1 层,我们对层树为 $w(w>1)$ 的节点使用数学归纳法证明.

(1)当 $w=2$ 时,V_2 的父节点是根节点 rt,我们用反证法证明这种情况下的命题成立. 假设存在一个对象 p,$p\in\nabla(V_2)$,但 $p\notin CA(V_2)(RP(V_2))$. 因为 p 是 V_2 上的 skyline 对象,所以不存在对象 r,使得 r 在 V_2 上支配 p,那么存在两种情况:①不存在任何对象 a,使得 a 在 V_2 上的取值均与 p 相同,并且 a 在 $rt\text{-}V_2$ 上支配 p;此时我们知,p 为根节点 rt 上的 skyline 对象,所以 $p\in CA(V_2)$,这与 $p\notin CA(V_2)\cup RP(V_2)$ 矛盾. 因此,假设不成立. ②存在 β 个对象,使得这 β 个对象在 V_2 上的取值均与 p 相同,并且在 V_2 上均支配 p. 我们取这 β 个对象当中,在 $rt\text{-}V_2$ 上不为其它对象支配的对象,记为 b,那么不难看出,对象 b 是 V_2 上的 skyline 对象,同时又是根节点 rt 上 skyline 对象,因此 $b\in CA(V_2)$;又因为 b 在 V_2 上的取值均与 p 相同,所以 $p\in RP(V_2)$,这与 $p\notin CA(V_2)\cup RP(V_2)$ 矛盾. 因此,假设不成立. 根据以上的分析,我们可知,这种情况下命题成立.

(2)假设 $w=n$ 时,第 n 层的节点 V_n 满足:$\nabla(V_n)=CA(V_n)\cup RP(V_n)$.

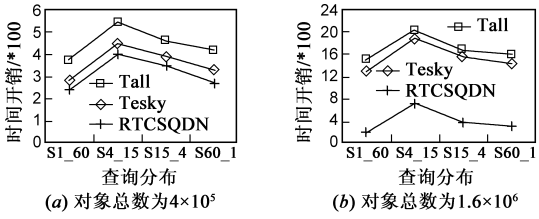
(3)当 $w=n+1$ 时,我们将证明 $\nabla(V_{n+1})=CA(V_{n+1})\cup RP(V_{n+1})$ 成立. 假定第 $n+1(n>1)$ 层节点 V_{n+1} 的父节点为 V_n,而 V_n 的父节点为 V_{n-1},因为 $CA(V_n)\subseteq CA(V_{n-1})$,所以,$AD\text{-}CA(V_{n-1})\subseteq AD\text{-}CA(V_n)$,因此有,$RP(V_n)\subseteq AD\text{-}CA(V_n)$. 从而,我们可以将 $RP(V_{n+1})$ 中对象分为两部分:① $FP=RP(V_{n+1})\cap RP(V_n)$,以及② $SP=RP(V_{n+1})\text{-}FP$,所以,$CA(V_{n+1})\cup FP$ 为使用 SFS 算法[3]从 $\nabla(V_n)$ 中获取的 V_{n+1} 上 skyline 对象,而 SP 为 $AD\text{-}\nabla(V_n)$ 中,在 V_{n+1} 上的取值与 $CA(V_{n+1})\cup FP$ 内某一对象均相同的对象组成的集合,因此根据(1)中的分析可知,$\nabla(V_{n+1})=CA(V_{n+1})\cup FP\cup SP=CA(V_{n+1})\cup RP(V_{n+1})$. 因此,当 $w=n+1$ 时,命题成立.

4　实验评估

这一节评估 RTCSQDN 算法的有效性. 实验环境为:PⅢ 1.60G CPU、512M 内存、60G 硬盘;Windows XP 系统. 所有实验代码在 java 编译器中运行通过. 我们使用两类数据集,即独立分布数据集和反相关分布数据集[2]. 在数据集中,对象的个数分别取 4×10^5 和 1.6×10^6 两种情况.

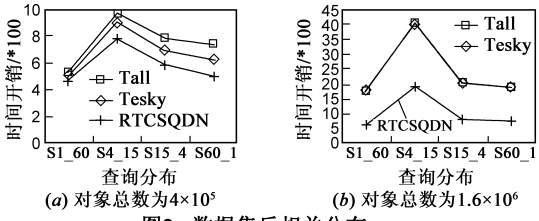
在实验中,与 RTCSQDN 算法比较的有两个方法:(1)文献[6,9,10]中使用的 Tall 方法;(2)文献[1]中使用 Tesky 方法. 实验中产生 60 个维空间上的 skyline 查询,具体种类描述如下:(a)产生 60 棵维空间树,每棵

维空间树只含有一个维数为 4 的维空间,记为 $S1\_60$;(*b*)产生 15 棵维空间树,每棵维空间树含有 4 个维空间,具体来说含有:维数为 6 的维空间 15 个、维数为 5 的维空间 18 个、维数为 4 的维空间 23 个以及维数为 3 的维空间 4 个,记为 $S4\_15$;(*c*)产生 4 棵维空间树,每棵维空间树含有 15 个维空间,具体来说含有:维数为 6 的维空间 4 个、维数为 5 的维空间 13 个、维数为 4 的维空间 16 个、维数为 3 的维空间 17 个以及维数为 2 的维空间 10 个,记为 $S15\_4$;(2)产生 1 棵包含 60 个维空间的维空间树,其中维数为 7 的维空间 1 个、维数为 6 的维空间 5 个、维数为 5 的维空间 11 个、维数为 4 的维空间 21 个、维数为 3 的维空间 9 个以及维数为 2 的维空间 13 个,记为 $S60\_1$. 此外,与文献[1]一样,网络带宽为 40kb/s. 实验结果如图 1 和图 2 所示.



(*a*) 对象总数为4×10^5 (*b*) 对象总数为1.6×10^6

图1 数据集独立分布



(*a*) 对象总数为4×10^5 (*b*) 对象总数为1.6×10^6

图2 数据集反相关分布

从图 1 和图 2 我们可以看出,本文给出的 RTCSQDN 算法在各种实验设置下均显著优于其余的 2 种方法,这主要因为以下两个原因:(1)RTCSQDN 算法产生的数据传输量最小,从而网络传输的时间开销最小;(2)三种方法引起的 CPU 计算的时间开销基本上一样. 同时,我们还发现当数据反相关分布时,Tesky 方法的总时间开销几乎等于 Tall 方法. 这是因为当数据反相关分布时,Tesky 方法的数据传输量接近于 Tall 方法. 例如,在图 2(*b*)的 $S4\_15$ 中,Tall 方法的总时间开销为 4080s,而 Tesky 方法的总时间开销为 4030s. 此时,RTCSQDN 算法的总时间开销为 1930s.

5 结论

网络传输开销是分布式 skyline 查询的性能瓶颈. 本文提出一种有效降低分布式 skyline 查询传输代价的三阶段方法 RTCSQDN. 为了平衡网络传输量和查询时间开销,我们将 RTCSQDN 算法第 1 阶段和第 3 阶段放在 LP 终端上执行,而将 CPU 敏感的第 2 阶段放在高性能的 HP 服务器上完成. 同时,在第 2 和第 3 阶段,我们给出一种新颖的多维对象传送策略 PTGPV 来最小化网络数据传输量,从而优化了网络传输的时间开销. 我们从理论上分析和证明本文算法的正确性. 此外,我们通过具体的实验验证了 RTCSQDN 算法的有效性和实用性.

参考文献:

[1] V Akrivi, D Christos, K Yannis, V Michalis. SKYPEER: Efficient subspace skyline computation over distributed data[A]. Proc IEEE ICDE'07[C]. Istanbul: IEEE Press, 2007. 416 – 425.

[2] K Deng, X Zhou, H Shen. Multi-source skyline query processing in road networks[A]. Proc IEEE ICDE'07[C]. Istanbul: IEEE Press, 2007. 796 – 805.

[3] J Chomicki, P Godfrey, J Gryz, D Liang. Skyline with presorting: theory and optimization[A]. Proc ICIIS'05[C]. Maryland: MIT Press, 2005. 216 – 225.

[4] I Bartolini, P Ciaccia, M Patella. Efficient sort-based skyline evaluation[J]. ACM Transaction on Database Systems, 2008, 33(4): 1632 – 1695.

[5] Y Tao, X Xiao, J Pei. SUBSKY: Efficient computation of skylines in subspaces[A]. Proc IEEE ICDE'06[C]. Atlanta: IEEE Press, 2006. 65 – 74.

[6] K Hose, C Lemke, K Sattler. Processing relaxed skylines in PDMS using distributed data summaries[A]. Proc ACM CIKM'06[C]. Arlington: IEEE Press, 2007. 425 – 434.

[7] J Lee, G You, S Hwang. Personalized top-k skyline queries in high-dimensional space[J]. Journal of Information Systems, 2009, 34(1): 45 – 61.

[8] M Sharifzadeh, C Shahabi. The spatial skyline queries[A]. Proc VLDB'06[C]. Seoul: VLDB Endowment, 2006. 751 – 762.

[9] S Wang, B Ooi, A Tung, L Xu. Efficient skyline Query processing on peer-to-peer networks[A]. Proc IEEE ICDE'07[C]. Istanbul: IEEE Press, 2007. 1126 – 1135.

(下转第 810 页)

作者简介:

黄震华 男,1980 年出生,博士,讲师,软件行业协会系统工程分会理事,CCF 会员. 主要研究方向为数据库、数据仓库、OLAP 分析、数据挖掘与知识发现等.
E-mail: jukie.huang@gmail.com

向 阳 男,1962 年出生,博士,教授. 主要研究方向为数据库、数据仓库、数据挖掘、决策支持系统与语义网等.

林 琛 女,1982 年出生,博士研究生,CCF 学生会员. 主要研究方向为数据库、数据挖掘、信息检索与社会网络分析等.

(上接第 852 页)

[10] P Wu, C Zhang, Y Feng, B Zhao, D Agrawal, A Abbadi. Parallelizing skyline queries for scalable distribution [A]. Proc EDBT'06[C]. Munich: Springer Verlag, 2006. 112 – 130.

[11] X Lin, Y Yuan, Q Zhang, Y Zhang. Selecting stars: the k most representative skyline operator[A]. Proc IEEE ICDE'07[C]. Istanbul: IEEE Press, 2007. 86 – 95.

[12] B Yang, H Molina. Designing a super-peer network[A]. Proc IEEE ICDE'03[C]. Bangalore: IEEE Press, 2003. 49 – 60.

[13] S Naouali, R Missaoui. Flexible query answering in data cubes [A]. Proc DAWAK'05 [C]. Copenhagen: Springer Verlag, 2003. 221 – 232.

[14] W Zhang, X Lin, Y Zhang, W Wang, J Yu. Probabilistic skyline operator over sliding windows[A]. Proc IEEE ICDE'09 [C]. Beijing: IEEE Press, 2009. 513 – 524.

[15] B Chen, R Ramakrishnan, K LeFevre. Privacy Skyline: Privacy with Multidimensional Adversarial Knowledge [A]. Proc VLDB'07[C]. Vienna: VLDB Endowment, 2007. 770 – 781.

[16] S Chaudhuri, N Dalvi, R Kaushik. Robust cardinality and cost estimation for skyline operator[A]. Proc IEEE ICDE'06[C]. Atlanta: IEEE Press, 2009. 64 – 73.