

多数据流上的连续分布式 Top-k 监测^{*})

邓波 饶翔 贾焰 杨树强

(国防科技大学计算机学院 长沙 410073)

摘要 近年来,分布式系统中的数据流监测是一个十分活跃的领域。研究了如何实现通用并且高效的分布式 top-k 监测,即在分布的多数据流中根据用户给定的排序函数连续监测最大的 k 个值。在实际应用中,用户给定的排序函数可能是任意的排序函数,然而,目前的分布式 top-k 监测技术只支持加法作为排序函数。提出了一种通用的支持任意的连续的严格单调的聚集函数的分布式 top-k 监测算法 GMR。GMR 的通讯代价和 k 无关。通过真实世界数据和模拟数据验证了 GMR 的效率。实验表明,GMR 的网络通讯量比同类方法低一个数量级以上。

关键词 GMR, 分布式, Top-k, 数据流, 监测

Continuous Distributed Top-k Monitoring over Data Streams

DENG Bo RAO Xiang JIA Yan YANG Shu-Qiang

(School of Computer Science, National University of Defense Technology, Changsha 410073)

Abstract Monitoring data streams in a distributed system is the focus of much research in recent years. This paper addresses the generic and efficient processing of distributed top-k monitoring, which is continuously reporting the k largest values according to a user-specified ranking function over distributed multiple data streams. In practice, the user-specified ranking function would be arbitrary ranking function. Unfortunately, state-of-art distributed top-k monitoring approaches only support the sum function as the ranking function. In this paper, we present a general algorithm GMR for distributed top-k monitoring, which supports arbitrary continuous and strict monotone aggregation functions. The communication cost of GMR is independent of k . We verify the effectiveness of GMR empirically using both real-world and synthetic data sets. We show that GMR reduces overall communication cost by an order of magnitude compared with alternatives.

Keywords GMR, Distributed, Top-k, Data streams, Monitoring

1 前言

top-k 查询的目标是根据用户定义的排序函数找出评价价值最高的前 k 个结果。在许多应用中, top-k 查询是一种非常自然的应用需求,例如在多媒体数据库中的相似性查询、Web 数据库检索、数据流和数据挖掘中,面对大量的数据,用户往往不关心所有返回的结果,而只关心很小一部分的结果。因此,文[9]提出 top-k 查询思想后,立即引起了国内外研究者广泛的关注^[12~14,16,20]。

近几年,数据流的在线监测是一个非常活跃的研究领域,在股票交易、网络流量监控、网络安全监控、电信通信记录、传感器网络等许多领域有着广泛的应用^[6,7,18]。数据流的检测常常需要监测产生于分布在不同地域的数据源,如分布式拒绝服务攻击 (detecting distributed denial-of-service attacks, DDoS) 的监测。

在分布的多数据流环境中,实现 top-k 在线监测的朴素方法是把各数据源的数据都发送到中央节点进行 top-k 查询,然而这种处理方法往往会超出监测系统的处理能力,包括存储、通讯、计算能力等等。文[4,5]讨论了如何在分布环境下实现有效的数据流的 top-k 在线监测,通过在远端数据源

建立被监测对象的约束关系,保证全局 top-k 集合也是各监测节点上的 top-k 集合,只有在约束被打破时才进行节点间通讯,这样有效地减少了 top-k 在线监测的网络通讯量。

在文[11]中,我们提出一种更为有效的分布式 top-k 算法 MR。和文[4,5]中的方法相比,MR 显著地减少了监测过程中的网络通讯量,并且 MR 的通讯代价独立于 k 值。但是,这两种方法都只支持加法作为排序函数。而在实际应用中,用户给定的排序函数可能是任意的排序函数。

在本文中,我们提出了一种通用的支持任意的连续的严格单调的聚集函数的分布式 top-k 监测算法 GMR。GMR 通过对远程数据流建立约束关系来维护 top-k 结果集。GMR 只有在约束被打破时才需要节点间通讯,并且通讯时只需要传输少量的对象,通讯代价和 k 无关。通过真实世界数据和模拟数据验证了 GMR 的效率。实验表明,GMR 的网络通讯量比同类算法低一个数量级以上。本文的主要创新点如下:

- 1) 给出了在分布的多数据流上的 top-k 监测的形式化模型;
- 2) 提出了一种支持任意的连续的严格单调的聚集函数的分布式 top-k 监测算法 GMR;
- 3) 通过大量的实验验证了 GMR 的效率。

^{*}) 基金项目: 国家“九七三”重点基础研究发展规划基金项目 (2005CB321804); 国家“八六三”高技术研究发展计划基金项目 (2004A112020); 国家“八六三”高技术研究发展计划基金项目 (2005AA112030)。邓波 博士生, 主要研究方向为分布式计算, 数据库; 饶翔 硕士生, 主要研究方向为数据库, 海量信息处理。贾焰 博士, 教授, 博士生导师, 主要研究方向为分布式计算, 数据库等; 杨树强 博士, 教授, 主要研究方向为数据库, 海量信息处理等。

本文第2节给出了分布式 top-k 监测模型;第3节讨论了分布式 top-k 监测中的约束关系;第4节提出了支持任意的连续的严格单调的聚集函数的分布式 top-k 监测算法 GMR 算法;第5节是实验;第6节讨论了相关工作;最后对本文的工作进行总结。

2 问题描述

本节给出分布式 top-k 监测模型。本文关注的问题是在分布的多数据流中根据用户定义的排序函数连续监测最大的 k 个值。本文关注的是一个有 $m+1$ 个节点的分布式在线监测系统:一个中央协调节点 N_0 和 m 个远端监测节点 $N_1, N_2, \dots, N_m$ 。

定义1 给定一组远程监测节点 $N_1, N_2, \dots, N_m$, 以及当前监测对象集 $U = \{O_1, O_2, \dots, O_n\}$, 对被监测对象 $O_i \in U$, 令 $V_{i,r}$ 为 O_i 在 N_r 上的监测值。

定义2 令 V_i 为对象 $O_i \in U$ 在中央协调节点 N_0 上的聚集值, 则

$$V_i = f(V_{i,1}, V_{i,2}, \dots, V_{i,m}) \quad (1)$$

其中, f 是用户给定的排序函数。

本文只考虑 f 是连续的严格单调函数的情况。为简单起见, 只讨论 f 是连续的严格单调递增函数的情况。若 f 是连续的严格单调递减函数, 各监测节点只需监测 $-V_{i,r}$ 即可。为简单起见, 假定监测值和聚集值都在实数域, 并且 $V_{i,r} \geq 0$ 。注意到在监测中, U 中所有对象可能不会被所有监测节点监测。当对象 O_i 在当前窗口中没有被监测节点 N_j 监测时, 则认为 $V_{i,r} = 0$, 这并不影响定义1的普适性。

定义3 分布式 top-k 监测是连续的找出集合 $T \subseteq U$ 满足: $|T| = k$ 且 $\forall O_i \in T, \forall O_j \in U - T$, 有

$$V_i \geq V_j \quad (2)$$

其中, k 为给定的 top-k 监测结果集大小。

在实际系统中, 当 T 发生变化时, 系统需要重新确定新的 top-k 结果集 T' , 在这一处理过程中, 系统继续把 T 作为当前的 top-k 集合, 这与真实的 top-k 集合 T' 不一致, 但这一处理过程所需时间很短, 因此忽略不计。更详细的一致性讨论参见文[5]。

3 约束

一种朴素的支持任意排序函数的分布式 top-k 监测方法 (Naive Distributed Top-k Monitoring, 简称 NDTM) 是把各个数据流的数据都传送到 N_0 , 由 N_0 确定 top-k 结果集, 但是 NDTM 会引发大量的网络通讯。为了降低网络通讯量, 在本文中, 我们通过对远程数据流建立约束关系来维护 top-k 结果集。建立约束的基本思想是让全局最优的 k 个对象在各监测节点上也是最优的 k 个对象。只有在约束被打破时才需要节点间通讯。本节介绍 top-k 监测中的约束关系。

令 $\delta_{i,r}$ 为 U 中的每个对象 O_i 在监测节点 N_r ($1 \leq r \leq m$ 和中央协调节点 N_0 的调整因子, 令 $T \subseteq U$ 且 $|T| = k$ 。

定理1 对于任意的 top-k 结果集的都存在满足下列约束两个条件的调整因子, 并且满足下述两个条件的集合 T 是 top-k 结果集。

约束条件1: $\forall O_i \in U$, 有 $f(V_{i,1} + \delta_{i,1}, V_{i,2} + \delta_{i,2}, \dots, V_{i,m} + \delta_{i,m}) = V_i$;

约束条件2: $\forall N_r$ ($1 \leq r \leq m$), $\forall O_i \in T, \forall O_j \in U - T$, 都有 $V_{i,r} + \delta_{i,r} \geq V_{j,r} + \delta_{j,r}$ 。

证明: 首先证明对于任意的 top-k 结果集的都存在满足

上述约束两个条件的调整因子。把 U 中对象按聚集值的降序排列, 假定第 k 位的对象为 O_k , 由于 f 是连续的严格单调递增函数, 对于 $\forall O_i \in T$, 都可以找到 $\mu_i \geq 0$ 使得 $V_i = f(V_{k,1} + \mu_i, V_{k,2} + \mu_i, \dots, V_{k,m} + \mu_i)$, 对于 $\forall O_j \in U - T$, 都可以找到 $\mu_j \leq 0$ 使得 $V_j = f(V_{k,1} + \mu_j, V_{k,2} + \mu_j, \dots, V_{k,m} + \mu_j)$ 令 $\delta_{i,r} = V_{k,2} + \mu_i - V_{i,r}$ ($1 \leq j \leq m$), 则这样的调整因子满足上述两个条件。

其次证明满足上述两个条件的集合 T 是 top-k 结果集。由约束条件2知, $\forall N_r$ ($1 \leq r \leq m$), $\forall O_i \in T, \forall O_j \in U - T$, 都有 $V_{i,r} + \delta_{i,r} \geq V_{j,r} + \delta_{j,r}$ 。由于 f 是连续的严格单调递增函数, 即如果对于任意的 r ($1 \leq r \leq m$) 都有 $x_r \geq x'_r$, 则 $f(x_1, x_2, \dots, x_m) \geq f(x'_1, x'_2, \dots, x'_m)$ 。因此, 由约束条件1, $\forall O_i \in T, \forall O_j \in U - T$, 有 $V_i = f(V_{i,1} + \delta_{i,1}, V_{i,2} + \delta_{i,2}, \dots, V_{i,m} + \delta_{i,m}) \geq f(V_{j,1} + \delta_{j,1}, V_{j,2} + \delta_{j,2}, \dots, V_{j,m} + \delta_{j,m}) = V_j$ 。

由定义3知, 是 top-k 结果集。证毕。

4 分布式 Top-k 监测

本节给出支持任意的连续的严格单调的聚集函数的分布式 top-k 监测的通用最小刷新 (Generic Minimal Refresh, 简称 GMR) 算法。由定理1知, 在 top-k 监测中我们只需要连续的找出满足上述两个约束条件的集合就可以。

在初始化时, 我们可以通过 TA 算法^[13] 获得最初的 top-k 结果集, 然后采用的调整因子设置函数 Allocation (见算法1) 为各对象设置出满足约束条件的调整因子。

在监测过程中, 当监测节点 N_c 的约束被打破时, 令 $C \subseteq U$ 是监测节点 N_c 上打破约束的对象集, 称为冲突集, 令 $C_T = T \cap C$, 称 C_T 为向下冲突集, 令 $C_i = (U - T) \cap C$, 称 C_i 为向上冲突集。令 I 为 C 中聚集值较高的 $|C_T|$ 个对象的集合, 即 $I \subset C$ 且 $|I| = |C_T|$, $\forall O_i \in I, \forall O_j \in C - I: V_i \geq V_j$ 。在文[11]中, 我们证明了 $T' = (T - C_T) \cup I$ 是 top-k 结果集, 并且在重建约束时 N_c 需要传输的最小数据集为 C 和 C 在 N_c 上的监测值, 以及两个边界值:

$$B_c = \min_{O_i \in T} \{V_{i,c} + \delta_{i,c}\}, \max_{O_j \in U - T - C} \{V_{j,c} + \delta_{j,c}\}$$

$$H_c = \min_{O_i \in T} \{V_{i,c} + \delta_{i,c}\}, \max_{O_j \in U - T} \{V_{j,c} + \delta_{j,c}\}$$

但是在文[11]中, 在重建约束时, 调整因子的设置只支持 f 是加法的情况。

下面给出 GMR 算法的详细描述。

算法1 通用最小刷新算法 (Generic Minimal Refresh, 简称 GMR)

第1阶段: N_c 向 N_0 发送重建约束请求, 请求中包含冲突集 C 和 C 中对象在 N_c 的监测值, 以及两个边界值 B_c 和 H_c ;

第2阶段: N_0 从各节点 N_r ($r \neq c$) 分别获取冲突集 C 和 C 中对象在 N_r 的监测值, 以及两个边界值 B_r 和 H_r , 然后计算 C 中对象的聚集值。计算集合 I , 调用函数 Allocation 重新分配满足约束条件的调整因子, 更新 top-k 结果集为 $T' = (T - C_T) \cup I$, 并发送到所有节点。

函数 Allocation.

输入: $C, N, \{B_r\}, \{H_r\}, \{V_{i,r}\}, \{\delta_{i,r}\}$

输出: $\{\delta'_{i,r}\}$

1. 计算冲突集 C 中的每个对象 O_i 的聚集值

$$V_i = f(V_{i,1}, V_{i,2}, \dots, V_{i,m})$$

2. 计算冲突集 C 中的每个对象 O_i 的插值因子 α_i , 使得:

$$V_i = f(\alpha_i(H_1 - B_1) + B_1, \alpha_i(H_2 - B_2) + B_2, \dots, \alpha_i(H_m - B_m) + B_m)$$

3. 计算冲突集 C 中的每个对象 O_i 在各监测节点 N_r 上的调整因子:

$$\delta'_{i,r} = \alpha_i(H_r - B_r) + B_r - V_{i,r}$$

由于 f 是连续的严格单调递增函数, 因此 $\forall O_i \in C$, 都可以通过解析或迭代方法找到一个 α_i 满足 Allocation 的第 2 行的条件(迭代方法只要找到满足约束条件的近似解即可, 由中央协调节点记录代入 α_i 之后的 V'_i 与 V_i 的差值, 由于篇幅关系, 在这里不做详细的讨论, 假定 α_i 是精确解)。

下面证明算法 GMR 的正确性, 即 Allocation 生成的调整因子满足第 3 节所述的两个约束条件, 并且我们将证明 GMR 的通讯代价与 k 无关。

定理 2 Allocation 函数生成的调整因子满足约束条件 1. 即 $\forall O_i \in U$, 有

$$f(V_{i,1} + \delta'_{i,1}, V_{i,2} + \delta'_{i,2}, \dots, V_{i,m} + \delta'_{i,m}) = V_i$$

证明: 由于 Allocation 函数只涉及 C 中各对象的调整因子, 我们只需要证明重新生成调整因子后的 C 中各对象的聚集值不变即可。由 Allocation 的第 2 行和第 3 行可知, 重新生成调整因子后聚集值不变。证毕。

定理 3 Allocation 函数生成的调整因子满足约束条件 2. 即 $\forall N_r (1 \leq r \leq m), \forall O_i \in T', \forall O_j \in U - T'$, 都有: $V_{i,r} + \delta'_{i,r} \geq V_{j,r} + \delta'_{j,r}$ 。

证明: 由于 $T' = (T - C_T) \cup I$, 下面分情况讨论:

1. $\forall O_i \in T - C_T, \forall O_j \in U - T - C_{U-T}$, 由于 Allocation 不改变 O_i 和 O_j 的调整因子, 并且 O_i 和 O_j 都没有打破约束, 因此仍然满足约束条件 2;

2. $\forall O_i \in I, \forall O_j \in C - I$, 由 I 的定义知, $V_i \geq V_j$, 由于 f 是严格单调递增函数, 根据 Allocation 函数的第 2 行和第 3 行可知, O_i 和 O_j 满足约束条件 2;

3. 令 $H = f(H_1, H_2, \dots, H_m)$, 令 $B = f(B_1, B_2, \dots, B_m)$, 由 H_r 和 B_r 的定义可知 $H_r \geq B_r$ 。 $\forall O_i \in T - C_T, \forall O_j \in C - I$, 由 I 的定义知, $V_i \geq H \geq B$, 由于 f 是严格单调递增函数, 因此 $\forall O_i \in C - I, \alpha_i \leq 1$, 即 $\forall O_i \in T - C_T, \forall O_j \in C - I, O_i$ 和 O_j 满足约束条件 2;

4. $\forall O_i \in I, \forall O_j \in U - T - C_{U-T}$, 由 I 的定义知, $V_i \geq B \geq V_j$, 由于 f 是严格单调递增函数, 因此 $\forall O_i \in I, \alpha_i \geq 0$, 即 $\forall O_i \in I, \forall O_j \in U - T - C_{U-T}, O_i$ 和 O_j 满足约束条件 2。

综上所述, $\forall O_i \in T', \forall O_j \in U - T', O_i$ 和 O_j 满足约束条件 2。证毕。

定理 4 GMR 的通讯代价与 k 无关。

证明: 只有当约束被打破时, GMR 才需要节点间通讯。在重建约束的过程中, GMR 只需要从各监测节点获取 C 中对象监测值和两个边界值, 并将 C 中对象新的调整因子发送到各监测节点。 C 中只包含发生冲突的对象, C 的大小与 k 无关, 因此 GMR 的通讯代价与 k 无关。证毕。

5 实验

据笔者所知, 到目前为止, 本文是第一次支持任意的连续的严格单调的聚集函数的分布式 top-k 监测的研究。因此, 在实验中, 我们将 GMR 和第 3 节所述的朴素的分布式 top-k 监测算法 NDTM 进行比较。性能的衡量指标是监测时的网络流量。

我们使用了多种真实世界数据和仿真数据来评估我们的

算法, 由于篇幅原因, 我们只列出两种数据集的结果, 分别是:

1) WorldCup: 我们采用 1998 年 6 月 15 号 FIFA 世界杯官方网站的 HTTP 服务器日志, 该数据集包含来自 29 个服务器对大约 17000 个网页的 5800 万次请求^[1]。

2) Zipf: 我们用服从 Zipf 分布^[22] 的数据来取代 WorldCup 数据集中的对各网页请求次数总和, 然后均匀分布到各节点。为简单起见, 我们取 Zipf 分布的因子为 0.8。

由于篇幅原因, 我们只讨论常见的加法和乘法及其混和运算的情况。我们用下列三种函数作为聚集函数:

1. $f_1(x) = x_1 \cdot x_2$, 其中 x_i 表示网页 x 在上在 N_i 长度为 1 个小时的滑动窗口上收到请求次数的和, 以下 x_i 的含义相同。 $f_1(x)$ 代表聚集函数为乘法的情况。

2. $f_2(x) = x_1 \cdot x_2 + 2x_3$ 。 $f_2(x)$ 代表聚集函数为乘法和加法混和运算时的情况。

3. $f_3(x) = (x_1 + x_2)/2 + (x_3 + x_4 + x_5)/3$ 。 $f_3(x)$ 代表聚集函数为均值之和的时候的情况。

实验床和相关算法的实现采用 GNU C++。所有实验在一台 CPU 为 Intel Pentium IV 2.4GHz 和内存为 512MB 机器上运行, 操作系统为 Red Hat Linux 9。实验结果如下所示。

图 1 和图 2 分别显示了 $k=20$ 时 WorldCup 数据集和 Zipf 数据集的实验结果。从图 1 和图 2 中我们可以看到, GMR 的网络通讯量比 NDTM 低一个数量级以上(纵坐标为对数坐标)。这是由于 GMR 只有在约束被打破时才需要节点间通讯, 并且通讯时只需要传输少量的对象, 而 NDTM 需要把所有数据都传送到 N_0 。

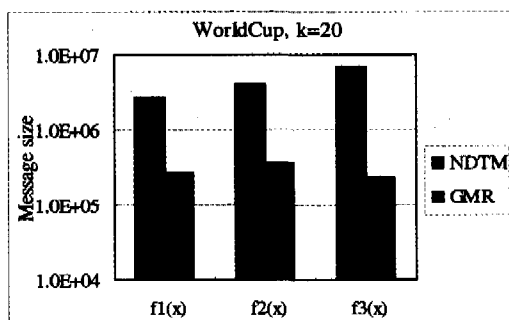


图 1 GMR 与 NDTM 在真实世界数据集上的性能比较

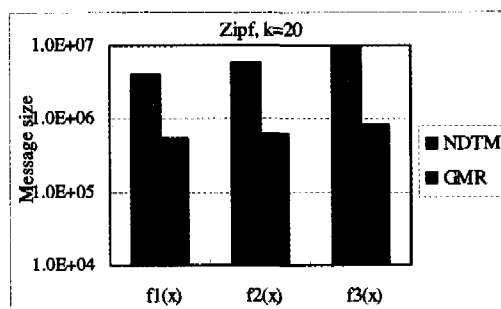


图 2 GMR 与 NDTM 在仿真数据集上的性能比较

我们验证 k 取不同值时 GMR 的性能, 图 3 显示了数据集为 WorldCup 时, 聚集函数为 $f_1(x)$ 时的实验结果(由于篇幅关系, 我们只列出数据集为 WorldCup, 聚集函数分别 $f_1(x)$ 时的结果, 其它结果是相似的)。从图 3 中我们可以看到, GMR 的网络通讯代价是与 k 无关的, 不随 k 的增长而增长。

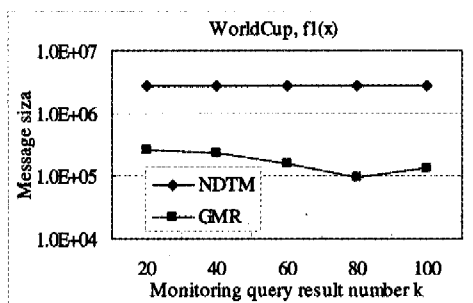


图3 $f_1(x)$ 在不同的结果集大小 k 上的性能比较

6 相关工作

最为著名的通用的 top-k 查询技术是由文[13, 14, 20]分别独立提出的阈值算法(Threshold Algorithm, 简称 TA),它是一种通用的适用于单一数据源的高效的 top-k 查询算法,但不适用于分布式系统。在 TA 的基础上,研究人员提出了许多面向分布式系统的 top-k 查询处理技术^[2, 3, 8, 10, 19],但这些算法都是快照式的,即一次完成,不能支持数据流上的连续查询。

文[4, 5]讨论了如何在分布环境下实现有效的数据流的 top-k 在线监测,通过在远端数据源建立被监测对象的约束关系,保证全局 top-k 集合也是各监测节点上的 top-k 集合,只有在约束被打破时才进行节点间通讯,这样有效地减少了 top-k 在线监测的网络通讯量。但文[4, 5]中的方法在每次约束被打破时都需要传输整个 top-k 集合。

在文[11]中,我们提出一种更为有效的分布式 top-k 算法 MR。MR 在约束被打破时只需要传输打破约束的对象集。因此,和文[4, 5]中的方法相比,MR 显著地减少了监测过程中的网络通讯量,并且 MR 的通讯代价独立于 k 值。但是,这两种方法假定排序函数为加法,都只是针对加法来建立和维护远端数据源的约束关系。而在实际应用中,用户给定的排序函数可能是任意的排序函数。本文提出的 GMR 算法可以支持任意的连续的严格单调聚集函数的分布式 top-k 监测,同时通讯代价和 k 无关。MR 算法可以看作是 GMR 算法的特例。

文[21]讨论了传感器网络中如何监测传感值最高的 k 个传感器的问题,并没有涉及聚集函数下的 top-k 监测。文[17]讨论了单个数据流上的多个聚集函数的 top-k 监测问题,着眼于如何减少单个节点上的 CPU 和内存使用,并没有考虑分布环境下如何减少网络通讯量。

总结 本文研究了如何实现通用并且高效的分布式 top-

k 监测,即在分布的多数据流中根据用户给定的排序函数连续监测最大的 k 个值。在实际应用中,用户给定的排序函数可能是任意的排序函数,然而,目前的分布式 top-k 监测技术只支持加法作为排序函数。在本文中,我们提出了一种通用的支持任意连续的严格单调的聚集函数的分布式 top-k 监测算法 GMR。GMR 通过对远程数据流建立约束关系来维护 top-k 结果集。GMR 只有在约束被打破时才需要节点间通讯,并且通讯时只需要传输少量的对象,通讯代价和 k 无关。通过真实世界数据和模拟数据验证了 GMR 的效率。实验表明,GMR 的网络通讯量比同类算法低一个数量级以上。如何在大规模分布环境下的实现通用的多 top-k 监测是我们下一步研究的目标。

参考文献

- 1 Arlitt M, Jin T. 1998 world cup web site access logs. August 1998. Available at: <http://www.acm.org/sigcomm/ITA/>
- 2 Bruno N, Gravano L, Marian A. Evaluating top-k queries over web-accessible databases. In: ICDE, 2002
- 3 Balke W-T, Nejdl W, Siberski W, et al. Progressive Distributed Top-k Retrieval in Peer-to-Peer Networks. In: ICDE, 2005
- 4 Babcock B, Olston C. Distributed Top-K Monitoring. In: SIGMOD, 2003
- 5 Babcock B, Olston C. Distributed top-k monitoring. [Technical report]. Stanford University Computer Science Department, 2002. <http://dbpubs.stanford.edu/pub/2002-61>
- 6 Carney D, Cetintemel U, Cherniack M, et al. Monitoring streams—a new class of data management applications. In: VLDB, 2002
- 7 Chen J, DeWitt D J, Tian F, et al. NiagaraCQ: A scalable continuous query system for internet databases. In: SIGMOD, 2000
- 8 Chang K C-C, Hwang S-W. Minimal probing: supporting expensive predicates for top-k queries. In: SIGMOD, 2002
- 9 Carey M J, Kossmann D. On saying “Enough already!” in SQL. In: SIGMOD, 1997
- 10 Cao P, Wang Z. Efficient top-k query calculation in distributed networks. In: PODC, 2004
- 11 Deng B, Jia Y, Yang S Q. Supporting Efficient Distributed Top-k Monitoring. To appear in WAIM, 2006
- 12 Fagin R. Combining fuzzy information from multiple systems. J. Comput. System Sci., 1999, 58: 83~99
- 13 Fagin R, Lotem A, Naor M. Optimal aggregation algorithms for middleware. In: PODS, 2001
- 14 Guntzer U, Balke W-T, Kiefling W. Optimizing multi-feature queries for image databases. In: VLDB, 2000
- 15 Gibbons P B, Matias Y. New sampling-based summary statistics for improving approximate query answers. In: SIGMOD, 1998
- 16 何盈捷, 王 珊, 杜小勇. 纯 Peer to Peer 环境下有效的 Top-k 查询. 软件学报, 2005, 16(4): 540~552
- 17 Mouratidis K, Bakiras S, Papadias D. Continuous Monitoring of Top-k Queries over Sliding Windows. To appear in SIGMOD, 2006
- 18 Madden S, Hellerstein J M, Shah M, et al. Continuously adaptive continuous queries over streams. In: SIGMOD, 2002
- 19 Michel S, Triantafillou P, Weikum G. KLEE: A Framework for Distributed Top-k Query Algorithms. In: VLDB, 2005
- 20 Nepal S, Ramakrishna M V. Query processing issues in image (mul-timedia) databases. In: ICDE, 1999
- 21 Wu M, Xu J, Tang X, et al. Monitoring Top-k Query in Wireless Sensor Networks. In: ICDE, 2006
- 22 Zipf G K. Human Behavior and the Principle of Least Effort. Addison-Wesley Press, 1949

(上接第 121 页)

结束语 基于多视图的信息系统体系结构描述方法是设计信息系统体系结构的一种有效的方法,能够规范体系结构设计,促进体系结构的交流和重用,对提高信息系统设计质量有较强的支持作用。本文深入研究了基于多视图的体系结构描述方法,为设计人员根据不同的背景和需求,构建适合系统开发的体系结构框架提供指导。

参考文献

- 1 C4ISR Architecture Working Group. C4ISR Architecture Framework Version 2.0 [R]. U. S., Department of Defense, 1997
- 2 IEEE Architecture Working Group. IEEE Recommended Practice

- for Architectural Description. IEEE Std 1471, Draft Version 3.0, USA, 1998
- 3 Zachman J A. A Framework for Information Systems Architecture [J]. IBM Systems Journal, 1987, 26(3): 276~292
- 4 Sowa J F, Zachman J A. Extending and Formalizing the Framework for Information Systems Architecture [J]. IBM Systems Journal, 1992, 31(3): 590~616
- 5 DoD Architecture Framework Working Group. DoD Architecture Framework Version 1.0 [R]. U. S. Department of Defense, 2003
- 6 Kruchten P B. Architectural Blueprints—The “4+1” View Model of Software Architecture [J]. IEEE Software, 1995, 12(6): 42~50
- 7 张维明. 信息系统集成技术[M]. 北京: 电子工业出版社, 2002
- 8 Chief Information Officers (CIO) Council. Federal Enterprise Architecture Framework, v 1 [R/OL]. U. S. CIO Council, 1999.
9. <http://www.itpolicy.gsa.gov/mke/archplus/projectmodel-subjects/main-theciocouncilfeaver11.html>