面向内容发布订阅系统的混合事件匹配算法

尤 涛,杨 凯,杜承烈,钟 冬,朱怡安
(西北工业大学计算机学院,陕西西安 710129)

摘 要: 当前的事件匹配算法不能在高效匹配的同时满足频繁订阅变更的要求.在结合已有谓词索引算法和覆盖网络算法的基础上,融合谓词索引结构的易变更和覆盖网络的高效匹配特点,提出一种混合的事件匹配算法.算法将部分订阅覆盖关系从覆盖网络中剥离,以同谓偏序的形式引入到谓词索引结构中去,达到高效匹配的同时保留了谓词索引的易变更结构.实验表明,与同类算法相比该算法能够在频繁订阅情况下提供高效的匹配,从而满足相关应用的需求.

关键词: 内容发布订阅系统;事件匹配算法;谓词索引;覆盖网络;同谓偏序订阅
中图分类号: TP393 **文献标识码:** A **文章编号:** 0372-2112 (2015)02-0358-07
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2015.02.023

Hybrid Event Matching Algorithm for Content-Based Publish/Subscribe System

YOU Tao, YANG Kai, DU Cheng-lie, ZHONG Dong, ZHU Yi-an
(*Department of Computer Science and Engineering, Northwestern Polytechnical University, Xi'an, Shaanxi* 710129, *China*)

Abstract: Current typical content-based publish/subscribe systems are not efficient in subscription processing or event matching. This paper presents hybrid event matching algorithm (HEMA), a novel publish/subscribe systems which joins predicate indexing and testing network approaches. We put partially ordered subscription with same predicates, which are separated from testing network structures, into predicate indexing mechanism to sustain efficient matching, whilst changing large number of subscriptions. Finally, experiments and performance analysis show that HEMA significantly improve throughput of event propagation and reduce response time to subscription updates meanwhile.

Key words: content-based publish/subscribe; event matching algorithm; predicate indexing; testing network; partially ordered subscription with same predicates

1 引言

当前应用广泛的内容发布/订阅系统[1]中,允许订阅者在事件的内容上指定约束条件,具有订阅灵活和表达能力较强的特点,当用户发布一个事件时,系统需要将该事件与订阅中的每个约束条件进行匹配.已有的相关研究中,核心问题都是解决事件代理采用何种算法实现事件和大量订阅者之间的高效匹配[2].

然而,伴随着系统应用领域和范围的不断扩大,诞生了对内容发布/订阅的新需求.例如当前比较流行的算法交易(high frequency algorithmic trading, HFT)[3],应用了线性回归、博弈论、神经网络、遗传算法等对交易过程的订阅细节进行干预,这就带来了对订阅的大量动态变更.因此交易系统中就要求在满足高效匹配的前提下能够应对用户频繁更改查询(订阅)条件的需求,否则将给客户带来不可估量的损失.又如分布式虚拟环境[4](distributed virtual environment, DVE)中,交互信息量非常大,节点间关联关系复杂,随着试验过程的推进,各节点间频繁变更交互(订阅)关系.事件匹配与订阅变更的效率不仅决定了系统的实时性与可扩展性,而且会影响试验结果的真实性.

鉴于这些新应用对事件匹配算法的新要求,本文提出了一种混合事件匹配算法(hybrid event matching algorithm, HEMA),在频繁订阅情况下提供高效的匹配,以满足相关应用的需求.

2 相关工作

为了更好的对各类匹配算法进行分类,我们将能够反映算法主要特点的谓词索引、谓词关系、订阅覆盖作为坐标的三个维度进行建模,得到了如图1所示的算法

收稿日期:2013-10-09;修回日期:2014-02-28;责任编辑:蓝红杰
基金项目:国家863高技术研究发展计划(No.2011AA010102);国家自然科学基金(No.61303225);航空科学基金(No.2013553034)中央高校基本科研费专项基金(No.3102014)

分类.图中 x 轴为是否使用谓词索引(表)结构,使用 $x=1$不使用 $x=0$;y 轴为是否考虑谓词间关联,考虑 $y=1$不考虑 $y=0$;z 轴为是否考虑订阅覆盖,考虑 $z=1$ 不考虑 $z=0$.依据 $<x,y,z>$ 的不同取值,可以将各类匹配算法按如下方式分类:

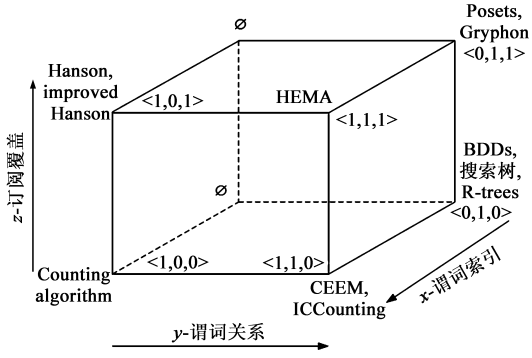


图1　事件匹配算法分类

(1) 计数类算法(Counting algorithm)[12],对应图中的 $<1,0,0>$ 位置.这类算法采取索引结构,易变更;未考虑谓词间关系、订阅覆盖等因素,匹配效率低.

(2)CEEM[6]算法、ICCounting 算法及其同类算法,对应图中 $<1,1,0>$ 的位置.这类算法也采取索引结构,易变更;未考虑订阅覆盖关系,匹配效率低.

(3)汉森算法(Hanson algorithm)及其扩展算法[8],占据了图中 $<1,0,1>$ 的位置.这类算法在索引结构上抽取了订阅覆盖关系,使得订阅结构不易变更;未考虑谓词间关系,匹配效率不高.

(4)基于覆盖网络的算法[9](Testing network based algorithm),对应图中 $<0,1,1>$ 的位置.这类算法能够充分利用各个订阅及谓词间的相关性,来提高匹配效率;受限于图的动态维护代价,算法往往只支持静态订阅或动态订阅效率处理效率不高.典型的有 Siena[7]的偏序集(partially ordered set,Poset)结构、Gryphon[5]系统的并行搜索树结构.

(5)基于匹配树的算法[10](Testing tree based algorithm),对应图中 $<0,1,0>$ 的位置.这类算法能够充分利用谓词间的相关性,实现较高的匹配效率;但受限于图的动态维护代价,算法往往只支持静态订阅.典型的匹配树算法有二叉决策图(binary decisions diagrams,BDD)及其改进算法、分布式 R-trees 的匹配算法[11]等.

通过上述分析不难发现,当前对匹配算法的研究已经相对成熟,只是高效的订阅结构(谓词索引结构)无法提供高效的事件匹配,而高效的事件匹配机制(覆盖网络、匹配树)无法提供高效的动态频繁订阅更改.即高效的订阅结构无法和高效的匹配机制很好结合,才造成了当前匹配算法无法直接支持频繁订阅类应用的情况.

而且,在已有研究中不难发现汉森算法及其扩展算法和 CEEM 算法都试图既采取谓词索引结构完成订阅,又应用谓词间与订阅间关系来进行匹配.只是 CEEM 算法采取了"欠融合"的方式,即在索引结构下没有充分利用订阅间的关联关系;汉森算法及其扩展算法又采取了"过融合"的方式,即在索引结构下考虑了过多的订阅间关联关系,使得索引结构实际演化成了覆盖网络结构,故而无法提供动态频繁订阅的支持.本文的 HEMA 分类情况如下:

(6)HEMA 占据了图中 $<1,1,1>$ 的位置,算法的谓词建立在索引结构上,在订阅过程中考虑了谓词间、订阅间的关联关系,详述见第三节.

3　HEMA

3.1　定义

定义 1　事件,谓词,订阅.事件 E 作为内容发布/订阅系统中用于系统内实体交互的单元,包含了属性 a 的序列 $[a_1,a_2,\cdots,a_n]$,这些属性都是基本数据类型.而事件实例 e 可被视作属性值 v 的一个记录 $[v_1,v_2,\cdots,v_n]$.基于典型的条件变量 $c::=\leqslant|<|=|>|\geqslant|\neq$,订阅谓词可以描述为 $P::=acv$ 的形式,即关于事件属性的一系列限制.由基本的谓词可以构成不同的订阅(条件)$\Phi::=\Phi\wedge P|\Phi\vee P|P$.

定义 2　谓词表.谓词分为上阈值谓词、下阈值谓词及等值谓词三类,具有相同属性和相同比较操作(上阈值、下阈值)的谓词归入一个谓词族;等值谓词存入哈希表中.每个谓词族对应谓词表的一行,并将阈值按序排列.此外,谓词表中将同一订阅中的谓词用指针链接起来,最后都指向订阅列表的具体订阅,如图 2 所示.

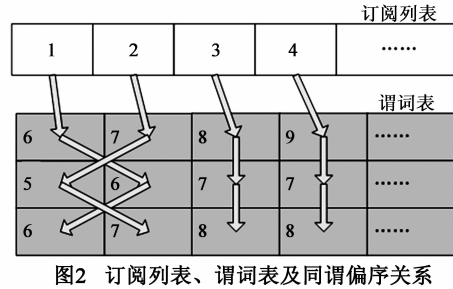


图2　订阅列表、谓词表及同谓偏序关系

定义 3　订阅列表.登记了所有加入谓词表的订阅,同时记录订阅在谓词表中第一个谓词的指针,以便于访问订阅中的所有谓词.

定义 4　同谓偏序订阅.指符合偏序关系且谓词相同的订阅.如图 2 所示,订阅 1 与订阅 2 虽然同谓,但由于存在交叉即不满足偏序,不存在同谓偏序关系;而订阅 1、订阅 3 同谓,由于不存在交叉即满足偏序,故存在同谓偏序关系.

定义 5 同谓偏序订阅列表.由同谓偏序订阅构成的订阅列表.图 2 中,订阅 1、3 构成的列表是一个同谓偏序订阅列表.

定义 6 最大同谓偏序订阅列表.在订阅列表中,满足同谓偏序关系数量最多的一组订阅.图 2 中,订阅 1、3、4 组成了最大同谓偏序订阅列表.

定义 7 最大 K 同谓偏序订阅列表.具有最大同谓偏序个数的 K 组订阅,形成的订阅列表.在实际应用中,可以设置门限值,即大于某个数的同谓偏序订阅构成一个列表.

3.2 数据结构

HEMA 算法的数据结构如图 3 所示,主要由同谓偏序订阅列表、混合订阅列表及其对应的谓词表(包括哈希表)组成.

同谓偏序订阅列表 P-list:表中存放的是最大 K 同谓偏序关系订阅.匹配时如果同谓偏序列表中的某订阅符合匹配关系,则其后继以及后继的后继也都符合匹配关系,可以大大减小遍历谓词表的规模.每个同谓偏序列表的谓词表(P-table)是单独存在的.图 3 右存放的就是"$x<$"、"$y<$"、"$z<$"三组谓词的同谓偏序订阅列表,它包含了"$x<, y<, z<$"、"$x<, y<$"、"$y<, z<$"、"$x<, z<$"、"$x<$"、"$y<$"、"$z<$"七个情形.显然,"$x<$"、"$y<$"、"$z<$"对应的同谓偏序列表只有一组,依序排列."$x<, y<, z<$"、"$x<, y<$"、"$y<, z<$"以及"$x<, z<$"对应的同谓偏序列表可能不止一组.另外,依据 K 的选择,某些谓词组也有可能不存在同谓偏序列表.

混合订阅列表 M-list:订阅混合列表中存放的是无法加入同谓偏序列表的订阅.哈希表无法映射为订阅偏序关系,因而归结到 *M-list* 中(图 3 左).

匹配界限 M-boundary:在 M-list 中,为了加快匹配速度,我们引入了匹配界限的概念.匹配界限是根据事件中所有属性的值来确定的.由于混合谓词表(M-table)中的单行数据是按照偏序关系排列的,所以当匹配界限确定后,只有所有谓词都在匹配界限内的订阅才会符合匹配,如果有一个或者多个谓词都超出了匹配界限,则该订阅无法与该界限对应的事件进行匹配.超出的形式可以是左右的超出,即超出谓词限定范围,如图 3 左的订阅 5;也可以是上下的超出,即超出了谓词范围,如图 3 左的订阅 1.

从 HEMA 算法的数据结构组织不难发现,首先,HEMA 采用的是索引结构;其次,谓词表与订阅列表的引入,使得匹配过程能够依照一定的谓词顺序进行,即考虑了谓词间的关联;最后,通过同谓偏序关系的引入,使得匹配过程依赖了部分的订阅覆盖,解决了"欠融合"问题,同谓偏序关系可以方便的从谓词表获得,也就避免了"过融合"问题.

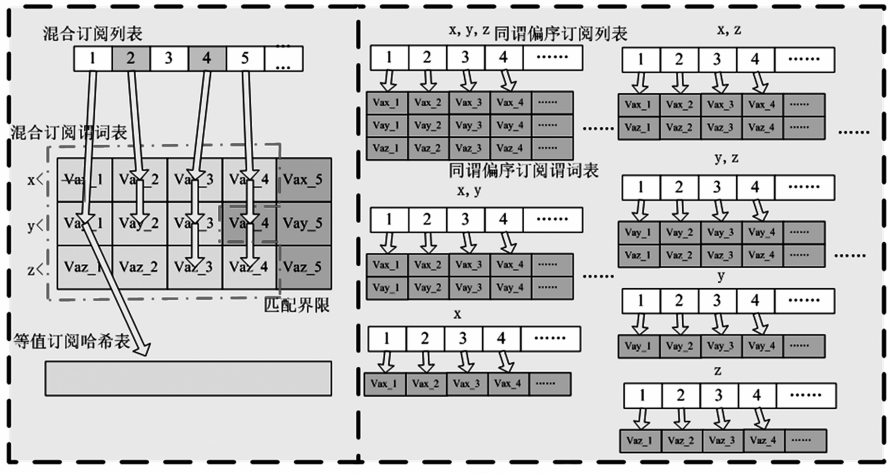


图3 HEMA数据结构

3.3 订阅流程

当接收到新的订阅后,需要对各主要数据结构分别进行更新,主要可分以下 3 步,如图 4 所示.

Step1:判断是否可加入 P-list,并插入.在对应的 P-list 中,找到距离该订阅最近的 precursor(前驱)和 successor(后继),如果该订阅的 precursor 和 successor 是直接相连的,或者 precursor,successor 有一个不存在,则说明其是可插入的,将该订阅插入 P-table,跳至 Step3.如果 precursor 或 successor 都不存在或不是直接相连,将该订阅加入 M-list.

Step2:在 M-list 对应的谓词表中插入该订阅.按照从上往下的顺序,在订阅所包含的每个谓词族中通过二分法查找相应位置,并将对应的阈值插入表 M-table 中;在谓词表插入的过程中,将插入的谓词按从上至下的顺序逐一进行链接,最后指向 M-list 中的对应位置.

Step3:标记开始和结束节点.

当用户取消订阅后,也需要对各主要数据结构分别进行更新,过程也是 3 步,如图 4 所示.

Step1:在 *P*-list 和 *M*-list 分别查找需要删除的订阅,如果在 *P*-list 中找到,进行 Step2,如果在 *M*-list 中找到,则直接进行 Step3.

Step2:将该订阅的 precursor 和 successor 取消与该订阅的连接,并将该订阅的 precursor 和 successor 连接起来.

Step3:从头结点开始,按照节点的连接顺序依次在谓词表 *P*-table、*M*-table 中删除节点.删除 *P*-list 或 *M*-list 中的订阅.

```
Subscribe(Subs)
1. NP = FindPrecursor(s)
2. NS = FindSuccessor(s)
3. If NP is NS direct precursor
4.    Add s between NP and NS
5.    Insert(s, P-table)
6. Else if NP not exists
7.    Add s before NS
8.    Insert(s, P-table)
9. Else if SP not exists
10.   Add s after NS
11.   Insert(s, P-table)
12. Else if NP and NS both not exists
13.   Add s into M-list
14.   Insert(s, M-table)
15. Return

Unsubscribe(Sub s)
1.  If find s in P-list
2.    Link precursor and successor of s
3.  n = s.StartNode
4.  While n! = NULL
5.     m = n.next
6.     Delete n
7.     n = m
8.  End while
```

```
Insert(Sub s, Table t)
1. Add start sign
2. For each predicate p of s do
3.    AddtoPredicateTable(p, t)//
二分查找并插入谓词表
4. Add end sign
5. Return

FindPrecursor(Sub s)
1. Find the node n which least big-
ger than s.StartNode in P-table
2. For each predicate p in s
3.    If n.value < p.value
4.       Return NULL
5.    n = n.next
6. Return n.subscription

FindSuccessor(Sub s)
1. Find the node n that most smaller
than s.StartNode in P-table
2. For each predicate p in s
3.    If n.value > p.value
4.       Return NULL
5.    n = n.next
6. Return n.subscription
```

图 4　订阅处理算法

3.4　匹配流程

设待匹配的订阅集合为 S_a,已匹配的订阅集合为 S_m.HEMA 算法的事件匹配分为五步,如图 5 所示.

Step1:在 *P*-list 找到带匹配事件所有子集的后继,如果能找到则将该后继以及其所有后继加入到匹配集 S_m 中.

Step2:分解事件,根据事件的中各个属性的值来确定匹配界限 *M*-boundary,将事件中所有属性对应的谓词表中带有开始标志且在匹配界限内的节点加入备选集 S_a 中.

Step3:更新 S_a,将 S_a 中的所有节点更新为其指向的下一个节点.如果新节点超出了 *M*-boundary,即不是事件中的匹配属性或者虽然是事件中的属性但不在 *M*-boundary 内,将该节点从 S_a 删除.

Step4:检测 S_a 中的节点是否带有结束标志,如果有的话将其对应的订阅加入到匹配集 S_m 中,并从 S_a 删除.

Step5:重复 Step3、Step4,直到 S_a 为空,此时 S_m 中的所有订阅即为匹配订阅.

```
Match(Event e)
1. Let S_m = empty
2. Let S_a = empty
3. For allsubset e' of e
4. NS = FindSuccessor(e')
5. While NS! = null
6.    Add NS into S_m
7.    NS = NS.Successor
8. End while
9. Make M-boundary from e
10. For each node n in M-boundary
11. If n.start = = true
12.    Add n into S_a
13. While S_a! = empty
14.    For each node n in S_a
15.       If n.end = = true
16.       Add n.subscription into S_m
17.       Delete n from S_a
18.    n = n.next
19.    If n is out of M-boundary
20.       Delete n from S_a
21. End while
22. Return S_m
```

图 5　事件匹配算法

从 HEMA 在订阅处理和事件匹配流程不难发现:(1)订阅的动态性,算法采用了谓词索引的结构存储订阅信息,且订阅的更新过程只与索引结构打交道,而索引结构是很容易完成更新的;(2)匹配的高效性,对谓词族进行了划分,并对每个谓词族用排序或者散列表的方式进行了索引,使得在对每个谓词进行匹配时能获得很高的效率;利用谓词间的关联性以及部分订阅间的覆盖关系(即同谓偏序覆盖关系),进一步加快匹配速度.

3.5　性能分析

假设支持的属性数目为 k,处理的总订阅数目为 n.

空间复杂度:算法的空间复杂度主要是谓词表的空间需求.每条订阅最多由 k 个谓词组成,在谓词表的谓词数是 kn,即总的空间复杂度是 $O(kn)$.

订阅插入混合列表的复杂度:设第 i 行的谓词数目为 P_i,插入一个谓词的复杂度为 $O(\log P_i)$,每个订阅最多由 $2k$ 个谓词构成,则插入一条订阅的复杂度为 $O(\log P_i \times 2k)$,最坏的情况是每一行都有 n 个谓词,则复杂度为 $O(k\log n)$.

订阅插入同谓偏序列表的复杂度:设同谓偏序列表中的订阅数量为 m,找到最近节点的复杂度为 $O(\log m)$,由于属性数目有限,判断偏序关系的复杂度可以认为是 $O(1)$,综合复杂度为 $O(k\log m)$.m 的最大值

为 n,即所有订阅都可存入同谓偏序列表,此时复杂度为 $O(k\log n)$.

而删除、更新与插入操作类似,因此复杂度均为 $O(k\log n)$.

混合订阅列表中事件匹配的复杂度:设事件 e 包含的属性数目为 j,设第 i 行的谓词数目为 P_i.确定匹配界限的复杂度为 $O(j\times\log P_i)$,设第 i 行落在匹配界限内的谓词数目为 Pin_i,寻找头结点的复杂度为 $O(j\times Pin_i)$,设符合条件的订阅数目为 l,其对应的谓词数目为 q,则确定是否匹配的复杂度为 $O(lq)$,最后总的复杂度为 $O(j\times(\log P_i+Pin_i)+lq)$,最坏的情况下,$j=k$,$l=n$,$q=k$,则最终的复杂度约为 $O(kn)$.

同谓偏序订阅列表中事件匹配的复杂度:设同谓偏序列表中的订阅数量为 m,在同谓偏序列表中找到事件位置的复杂度为 $O(\log m)$,找到位置之后,其后面的所有订阅可以直接匹配,因此,同谓偏序列表中的事件匹配的复杂度为 $O(\log m)$. m 的最大值为 n,即所有订阅都在同谓偏序列表中,此时复杂度为 $O(k\log n)$.

因此,事件匹配的复杂度取决于同谓偏序列表中订阅数量占总体订阅数量的比例,同谓偏序列表中的订阅越多,复杂度越小,极端最好情况即如果订阅都在同谓偏序列表中,则总体复杂度为 $O(k\log n)$,反之为 $O(kn)$.

表 1 HEMA 与典型索引计数、覆盖网络算法复杂度比较

算法	订阅变更复杂度	事件匹配复杂度
典型索引计数[12]	$O(k\log n)$	$O(kn)$
典型覆盖网络[7]	$O(kn)$	部分 $O(k\log n)$,部分 $O(kn)$
HEMA	$O(k\log n)$	部分 $O(k\log n)$,部分 $O(kn)$

表 1 为 HEMA 与典型索引计数、覆盖网络算法复杂度的比较.在订阅变更复杂度方面,HEMA 和典型索引计数算法均为 $O(k\log n)$,小于典型覆盖网络算法的 $O(kn)$.在事件匹配复杂度方面,HEMA 和典型覆盖网络算法相当,小于典型索引计数算法的 $O(kn)$.

4 实验与结果

4.1 实验设计

在基于索引结构的匹配算法中,我们选取当前最优的 BE-Tree 算法为基准[12].覆盖网络算法的研究有很多,但主要是针对其应用范围进行扩展[7],基本的算法核心和处理流程没有发生很大改变,因此我们选取了应用广泛的基于 Poset 的算法作为代表.

实验环境包括一台运行各匹配算法的路由服务器、3 台用于发布的主机和 6 台用于订阅的主机,百兆以太网环境.其中路由服务器配置为:CPU INTEL E8500 2.93GHz,RAM 2GB,Windows XP Professional.在典型的虚拟试验场景下,进行某武器系统的对抗试验,发布实体为若干战斗机,订阅实体为若干战术雷达.为了方便进行 HEMA 与同类算法的比较,在相同的环境下,设计了两组实验,实验参数值如表 2 所示.实验中的属性为飞机的六个自由度属性、位置属性、时间属性等,除时间属性是 int 类型外,其余为 double 类型.

表 2 实验的参数设置

参数	实验 1	实验 2
属性数	13	15
订阅中的谓词族数	1 – 6	3 – 8
等值谓词比例	10%	10%
同谓偏序关系比例	50%	70%

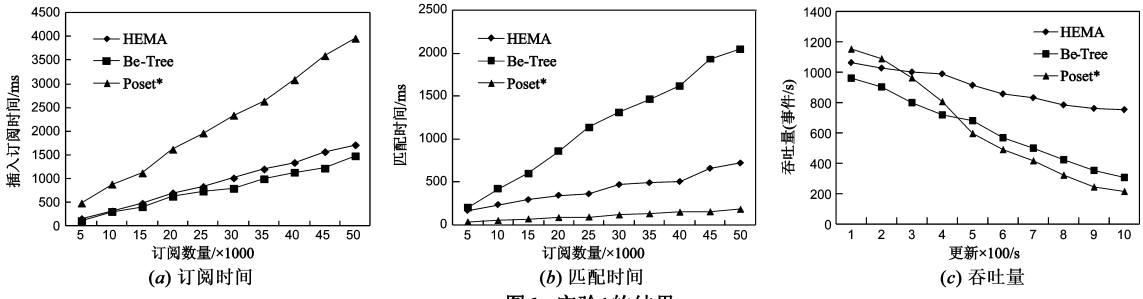
4.2 结果与分析

对两组实验,分别插入 50000 条订阅.每隔 5000 条订阅进行取样,记录插入 5000 条订阅和匹配 200 个事件所需的时间.图 6(a)(b)和图 7(a)(b)分别记录了两组实验的结果.可见,在订阅插入方面(图 6(a)和图 7(a)),HEMA 和 BE-Tree 算法性能类似,基本上随着订阅数量的增长呈线性增长趋势,且插入订阅的时间与订阅中的谓词数以及已插入的订阅数成正比.Poset 由于维护了大量的订阅覆盖信息,且随着订阅增加有可能出现 Poset 集整体调整、甚至重构的情况,因而处理效率低于 HEMA 和 BE-Tree 算法.在事件匹配的效率比较方面(如图 6(b)和图 7(b)所示),BE-Tree 算法和 HEMA 算法中的匹配时间也随着订阅数量的不断增加呈线性增长,但是由于 HEMA 算法利用了谓词间的相关性以及订阅间的(同谓偏序)覆盖关系进一步减少了不必要的比较操作,其斜率远远小于 BE-Tree 算法.随着数据结构中的订阅数目越来越多,不具备同谓偏序的订阅也会随之增多,HEMA 算法对于这些订阅会退化为计数算法,虽然 HEMA 与 Poset 复杂度类似,但 Poset 维护了所有的订阅覆盖信息,且考虑了谓词间的相关性,因而匹配效率高于 HEMA.这些结果和 3.5 节的复杂度分析情况基本吻合.

另外,我们测试了三算法在订阅变更同时进行事件匹配的效率.当订阅更新从 100 次/s 递增到 1000 次/s,三算法的事件匹配吞吐量如图 6(c)和图 7(c)所示.当订阅更新速度不高时,三类算法的匹配吞吐量相当,甚至还出现了 Poset 算法在一开始有较好表现的情况;随着订阅更新速度的增加,Poset 算法受困与低效的订阅更新,BE-Tree 算法受限与其匹配速度不高,均有明显的吞吐量下降,HEMA 算法由于结合了两者的优点,取得了较好的效果.HEMA 与其他两个算法相比,订阅更新速度越大,匹配吞吐量的这种优势也更加明显,最好的情况下吞吐量为其他算法的 4 ~ 5 倍.

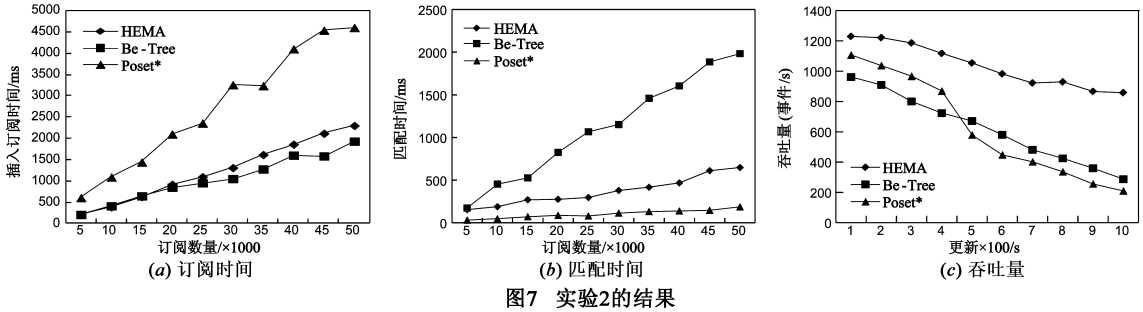
此外,通过比较可以进一步发现,在实验 2 中,HEMA算法的匹配效率优于实验 1.分析其原因,随着谓词数目的增加(从 1~6 个到 3~8 个),事件匹配的成功率将会随之降低,HEMA 算法中能够提前舍弃订阅的概率也将随着谓词数目的增加而增加;另一方面,人为控制同谓偏序列表的规模从之前的 50% 增加到了 70%,而 HEMA 算法在同谓偏序列表中的匹配复杂度为 $O(k\log n)$小于其在混合列表中的匹配复杂度 $O(kn)$,匹配效率自然有所提高.

可见,相对于其他算法,HEMA 更加适应于有大量动态变更订阅需求的应用.



(*a*) 订阅时间　(*b*) 匹配时间　(*c*) 吞吐量

图6　实验1的结果



(*a*) 订阅时间　(*b*) 匹配时间　(*c*) 吞吐量

图7　实验2的结果

5　结论

本文融合了谓词索引结构的易变更和覆盖网络的高效匹配特点,提出一种混合的事件匹配算法.算法将订阅覆盖关系从覆盖网络算法中剥离,并引入到谓词索引结构中去,进而达到了在订阅频繁增加、删除、变更情况下仍保持高效事件匹配的效果.实验表明,与同类算法相比该算法能够在频繁订阅情况下提供高效的匹配,从而能够满足频繁订阅类应用的需求.当然,仅仅优化事件匹配算法是不够的,下一步拟进行订阅表达方式、路由算法方面的研究,提高整个内容发布订阅系统的性能,以适应新的应用需求.

参考文献

[1] 刘殿兴,赵文,李信鹏,冯志明,张世琨,王立福.RFID 信息服务网络中支持复合订阅的路由算法研究[J].电子学报,2010,38(2):33-40.
Liu Dianxing, Zhao Wen, Li Xinpeng, Feng Zhiming, Zhang Shikun, Wang Lifu. Research on Routing Algorithm Supporting Composite Subscriptionsin RFID Information Service Network [J]. Acta Electronica Sinica, 2010, 38(2): 33-40. (in Chinese)

[2] 薛小平,张思东,张宏科,王小平,葛乐,尹琴.基于内容的发布订阅系统路由算法[J].电子学报,2008,6(5):953-961.
Xue Xiaoping, Zhang Sidong, Zhang Hongke, Wang Xiaoping, Ge Le, Yin Qin. Content based routing algorithms of the publish-subscribe systems[J]. Acta Electronica Sinica, 2008, 6(5): 953-961. (in Chinese)

[3] K.R. Jayaram, C. Jayalath, P Eugster. Parametric content-based publish/subscribe [J]. ACM Transactions on Computer Systems, 2013, 31(2): 44-88.

[4] Zengxiang Li, Xiaorong Li, TA Nguyen Binh Duong, Wentong Cai, Stephen John Turner. Accelerating optimistic HLA-based simulations in virtual execution environments[A]. Proceedings of the 2013 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation[C]. New York: ACM, 2013. 211-220.

[5] Ashayer G, Leung H K Y, Jacobsen H A. Predicate matching and subscription matching in publish/subscribe systems[A]. The 22nd International Conference on Distributed Computing Systems Workshops[C]. New York: ACM, 2002. 77-88.

[6] 陈继明,鞠时光,潘金贵,邹志文,龚震宇.基于内容的快速事件匹配算法[J].通信学报,2011,32(6):78-85.
Chen Jiming, Ju Shiguang, Pan Jingui, Zou Zhiwen, Gong

Zhenyu. Content-based effective event matching algorithm[J]. Journal on Communications, 2011, 32(6): 78 – 85. (in Chinese)

[7] Zigor Salvador, Aurkene Alzua, Mikel Larrea, Alberto Lafuente. Mobile XSiena: towards mobile publish/subscribe[A]. Proceedings of the Fourth ACM International Conference on Distributed Event-Based Systems[C]. New York: ACM 2010. 91 – 92.

[8] Shen Z H, Tirthapura S, Aluru S. Indexing for subscription covering in publish-subscribe systems[A]. IEEE International Conference on Data Engineering[C]. Piscataway: IEEE, 2005. 32 – 43.

[9] Kazemzadeh R S, Jacobsen H A. Opportunistic multipath forwarding in content-based publish/subscribe overlays[A]. Proceedings of the 13th International Middleware Conference[C]. New York: Springer-Verlag, 2012. 249 – 270.

[10] Anceaume E, Datta A K, Gradinariu M. A semantic overlay for self-peer-to-peer publish/subscribe[A]. 26th IEEE International Conference on Distributed Computing Systems[C]. Piscataway: IEEE, 2006. 22 – 26.

[11] Silvia B, Pascal F, Maria G. Content-based publish/subscribe using distributed R-trees[A]. International Conference on Parallel and Distributed Computing[C]. Berlin: Springer, 2007. 537 – 548.

[12] Mohammad Sadoghi, Hans Arno Jacobsen. BE-Tree: An index structure to efficiently match boolean expressions over high-dimensional discrete space[A]. 37th SIGMOD International Conference on Management of Data[C]. New York: ACM, 2011. 637 – 648.

作者简介

尤 涛 男,1983 年生于河南三门峡. 西北工业大学计算机学院讲师,研究方向为分布式事件系统与数据流处理.

E-mail: youtao@nwpu.edu.cn

杨 凯 男,1989 年生于河北沧州. 西北工业大学计算机学院硕士研究生,研究方向为数据流处理.