基于内容的发布/订阅模型中高效的匹配算法*

张彩云， 康亚男， 成汝震

(河北师范大学 数学与信息科学学院,河北 石家庄 050016)

摘 要:高效的匹配算法是大规模基于内容的发布订阅系统所要研究的热点问题之一.提出了一种快速有效的算法,算法根据逻辑表达式的特点,对所有订阅按照优先级进行预处理操作,使相同属性的比较次数小于等于 1 次,从而降低了匹配的代价.适合应用于大规模分布式基于内容的发布订阅系统中.

关键词:基于内容; 发布/订阅; 匹配算法; 优先级

中图分类号:TP 301.6 **文献标识码**:A **文章编号**:1000-5854(2009)04-0451-05

发布/订阅中间件系统具有异步、松耦合和多点通信的特点,在开发大规模分布系统中得到日益广泛的应用.发布订阅系统包括信息的生产者(发布者),信息的消费者(订阅者)以及事件代理.发布者发布信息(事件)给事件代理,订阅者向事件代理订阅感兴趣的事件,事件代理负责将接收的事件及时路由给感兴趣的订阅者[1].早期的发布/订阅系统有基于渠道和基于主题的 2 种.基于渠道的系统中订阅是通过指定渠道来实现的,订阅者接收所有发布到渠道中的事件,事件的接收与事件的内容无关.在这类系统中,订阅者不能指定它对渠道中的一部分事件感兴趣,表达能力很弱,但是实现简单.基于主题的系统中所有的事件按照主题划分成组,每个事件只属于其中一个主题.订阅者在订阅信息时,指明对哪个主题感兴趣,将来一旦出现该主题下的事件,系统都会自动将其发送给订阅者,订阅者接收其订阅主题的所有事件.这种系统不去理解事件的内部结构,而将其看成是一个封闭的黑盒子,订阅者不能更细粒度地表达他对某主题下的一部分事件感兴趣,其表达能力较弱.

针对基于渠道和主题系统的不足,现在提出了基于内容的发布/订阅系统,这种系统中,事件不再依赖于外部的某个标准(如渠道、主题等)分类,而是按照事件本身的内容分类.订阅者根据事件的内容来订阅事件,不必受系统预先定义的主题的限制.简单说,在基于内容的发布/订阅模型中,一条消息 M 被看成由一些(属性,值)对组成,例如一条有关网上拍卖信息的消息为(Author,Tom)、(Publisher,OReilly).而一条订阅 sub 被看成是对消息中若干属性值的一个布尔判断,如(Author =“Tom”)and(Publisher =“OReilly”).当且仅当 sub(M) = true 时,消息 M 与订阅 sub 匹配.该模型图形化表示见图 1.

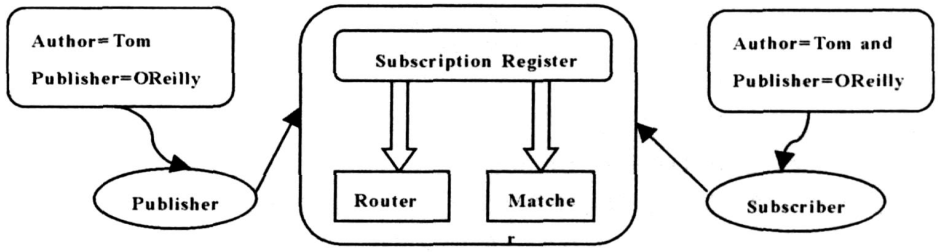


图 1 基于内容的分布式发布/订阅系统

基于内容的发布订阅系统更加灵活,但是也使系统的设计更加复杂化,其中一个核心问题就是如何实现

* **收稿日期**:2007-11-13; **修回日期**:2008-09-15
基金项目:河北省教育厅基金(2004361); 河北师范大学自然科学基金(120128)
作者简介:张彩云(1980-),女,河北张家口人,硕士研究生,研究方向为计算机网络及应用.

大量事件与大量订阅之间基于内容的高速匹配,即匹配算法[2].

1 传统算法

匹配算法的实质是:给定一条事件 e 和一个订阅的集合 S,找出满足 e 的所有的订阅. 事件 e 由一组(属性,值)对连接所构成,没有一个值对包含相同的属性. 如(username, Tom),(Number Of Orders,100),(Price, 200)就是一个事件. 而订阅是由一组谓词所构成,每个谓词由一个三元组所构成,包括属性、值和操作符,如订阅 s = (username = "Tom" and Number Of Orders > = 50 and Price < 200)就包含了 3 个谓词:(username, Tom, =),(Number Of Orders,50, > =),(Price,200, <). 事件 e 满足订阅 s 就是 s 中的每一个谓词都可以被事件 e 中的(属性,值)对所匹配,这里的事件 e 就不满足订阅 s,因为事件 e 中 Price = 200,谓词(Price, 200, <)没有被匹配.

最简单的匹配算法是将事件与每个订阅条件逐一相比较. 这种算法所耗费的时间与订阅者的数量密切相关. 基于内容的发布/订阅模型一般用于大规模的分布式系统中,消息发布量和订阅量都很大,当事件发布者以较快的频率发布各类事件时,对事件的匹配也就需要同样的快速,显然这种简单的逐一匹配订阅条件的算法无法满足这一要求. 研究人员提出了很多匹配算法如计数算法,匹配树算法及二叉判定图算法[3].

1) 计数法:该算法的基本思想是首先找出事件 e 满足的所有谓词 p;其次,根据第一步的结果判定 e 使 S 中哪些订阅条件满足. 定义 P 为包含所有谓词的列表,S 为包含所有订阅条件的列表,pred_to_subs 表示订阅条件和谓词之间的映射关系,nb_cond 表示每个订阅条件所包含的谓词数目,算法的输入 MatchingPreds 是事件 e 满足的所有谓词的集合. 为每个订阅条件设置 1 个计数器,遍历 MatchingPreds 中的谓词,通过映射表 pred_to_subs 查找它属于哪些订阅条件,并把订阅条件的计数器加 1. 最后,检查所有的订阅条件,当计数器的值与其包含的谓词数目相等时,该订阅条件被事件 e 匹配. 计数算法避免了同样的谓词被重复多次的计算,但最终由匹配的谓词集合计算出匹配的订阅条件的操作很耗时.

2) 匹配树:该算法的基本思想是将各订阅条件组织成一种树型结构,该树的深度为系统中的全部属性总数. 当一个事件到达时,系统按某一路径从树根到达某叶子节点,就得出了所有匹配成功的订阅条件. 该算法实际上是一种状态空间搜索方法. 该算法的时间复杂度较低,速度很快,但其空间复杂度为指数级. 同时,该算法的订阅维护的成本很高,每当客户增加订阅或取消订阅时,系统难以对该搜索树进行修改以反映订阅的变化,而必须要重建搜索树. 因此,该算法不太适用于大规模的系统.

3) 二叉判定图:该算法的基本思想是把订阅条件中的每个原子约束条件可以被看成是一个布尔变量,从而每个订阅条件可以用一个二叉判定图表示. 由于不同订阅条件中可能有相同的原子条件,则将各订阅的二叉判定图整合到一起,形成共享二叉判定图,其中的节点数为各订阅条件中的全部原子条件数量. 当一个事件到达时,先求出各原子条件的值,然后再通过对二叉判定图的遍历,以求出所有匹配成功的订阅条件. 该算法的优点在于它可以同时支持"与"、"或"2 种操作,而其他的订阅匹配算法基本上都是只支持原子约束条件之间的"与"操作,对于那些存在"或"操作的订阅条件,需要将其转化成多个只包含"与"操作的订阅条件来处理. 该算法的不足之处在于它依赖于生成二叉决策图时采取的变量顺序化策略,而如何选取最优顺序策略使得到的二叉决策图最简是一个 NP 完全问题.

2 一种高效的匹配算法

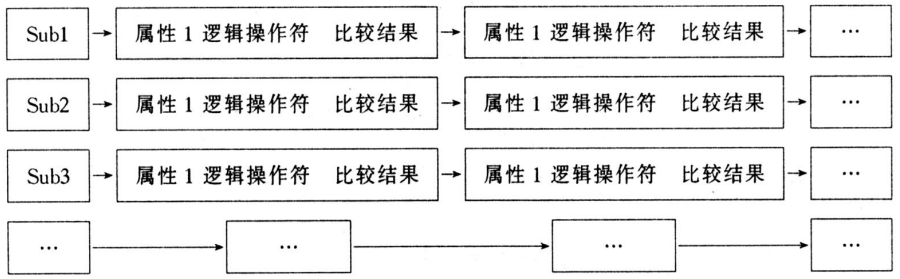
上面所述的基于内容的发布/订阅模式下的几种匹配算法的主要思想[4]就是在众多的消息中提取出它们相同的部分,减少对重复信息的处理,从而提高消息的匹配效率. 基于这种思想,本文中,笔者提出了新的匹配算法,该算法不仅使每个属性匹配的次数小于等于 1,而且同时支持"与"、"或"、"非"3 种逻辑操作.

2.1 数据结构

每一个订阅按分成的独立的属性单元组合成一个单向的链表 s,链表的每个点保存 3 个值:属性,逻辑操作符,匹配结果. 由链表 s 组成的订阅列表 S 见图 2.

建立一个属性列表 props,用来保存所有消息订阅者的属性表达式 prop,属性表见表 1.

图 2　订阅列表 S



2.2　算法的实现过程

表 1　属性列表 props

属性标号	属性值对	比较结果
1	Publisher = "OReilly"	true
2	Abstract LIKE "%JMS%"	false
…	…	…

1) 预处理. 遍历订阅条件,把每个订阅预处理后添加到订阅列表中 S. 预处理是指首先根据逻辑操作符 NOT, AND, OR 从高到低的优先级调换各属性的位置;然后逻辑操作符 AND 和它两端的属性看作一个处理对象,判断每一个逻辑操作符 AND 两端的运算符,看是否存在像 LIKE, BETWEENT, IN 等复杂度高的运算符,若只有一个这样的运算符并且在逻辑操作符的前面则把操作符前后的属性换位(注:本算法定义运算符 =, >, >=, <, <= 的复杂度相同,操作符 LIKE, BETWEENT, IN 的复杂度相同,但高于前面的运算符);最后将每个条件表达式分解成单个的属性表达式,每个属性表达式后面带上自己的逻辑运算符或结束标记,把每个订阅的属性表达式组织成一个订阅链表 sub.

2) 查阅属性列表匹配链表 sub 的属性. 遍历订阅列表 S 中的每个订阅链表 sub,对于订阅条件 sub 的每个属性 prop,查找属性列表 props(props 初始为空),若 prop 在 props 中存在,则查看它的匹配结果,把所查结果保存到订阅链表中,若不存在则将 prop 添加到 props 中,同时对该属性进行匹配,把将匹配结果保存在 props 中和订阅列表 subs 中.

3) 依据属性结果和逻辑值再匹配. 根据 2 匹配的属性结果的逻辑值,如果是 true,则判断跟在后面的逻辑操作符,倘若是 or,把它后面的属性值置为 true,倘若是 and,按照 2 继续下面的属性值匹配;如果是 false,同样判断跟在后面的逻辑操作符,若是 or,按照 2 继续下面的属性值匹配,若是 and,则直接把它后面的属性值置为 false. 依照这样的方法匹配,直到遇到结束标志符 #.

4) 计算链表 sub 的结果. 根据 3 的匹配结果来计算订阅链表的结果,判定该订阅条件是否匹配,若满足,将该订阅添加到匹配成功列表中.

2.3　实例描述

假设存在下面 4 条订阅:

Sub1: Author = "Jack" or Abstract LIKE "%JMS%" and Publisher = "QReilly" #

Sub2: Abstract LIKE "%JMS%" and Price < 100 or Type = "Computer" #

Sub3: Author = "Jack" or Publisher = "OReilly" and Type = "PDA" #

Sub4: Abstract LIKE "%JMS%" or Price < 100 or Publisher = "GReilly" #

可匹配的事件 e 为(Abstract, AJSMG), (Author, Jack), (Publisher, OReilly), (Price, 100), (Type, PDA). 对订阅进行预处理,处理结果如下:

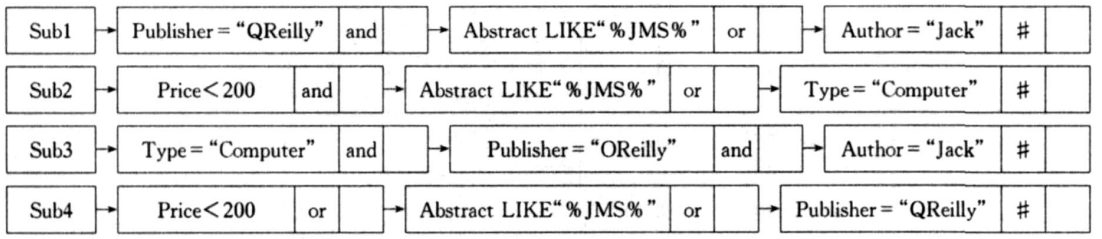
Sub1: Publisher = "QReilly" and Abstract LIKE "%JMS%" or Author = "Jack" #

Sub2: Price < 200 and Abstract LIKE "%JMS%" or Type = "Computer" #

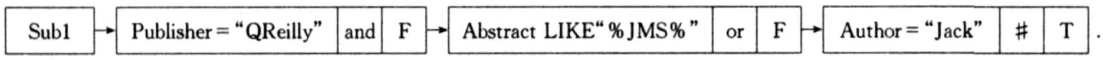
Sub3: Type = "Computer" and Publisher = "OReilly" or Author = "Jack" #

Sub4: Price < 200 or Abstract LIKE "%JMS%" or Publisher = "QReilly" #

or 然后划依据属性划分成独立的属性单元:



下面对属性进行匹配,逐一扫描订阅链表,对 Sub1 的匹配结果为



匹配 Sub1 的过程为:扫描 Sub1 时当碰到属性 Publisher = "QReilly"时(见表 2),首先查看属性列表 props(初始化为空)中是否存在该属性,如果不存在,那么把该属性添加到属性列表中,然后匹配该属性,匹配得结果为 F,把匹配得结果添加到属性列表中,然后查看该属性后面得逻辑操作符为 and 那么直接把该属性后面得属性直接设置为 F,再查看它后面得逻辑操作符为 or,则继续匹配该属性后面的属性 Author = "Jack",同样先查看属性列表 props 看该属性是否存在若没有,则把该属性添加到列表同时匹配该属性,匹配的结果为 T,把匹配的结果添加到属性列表中,查看它后面的操作符为结束标记 #,到此对 Sub1 的匹配过程完成,接着计算 Sub1 的结果为 T,匹配成功,把它添加到匹配列表中.继续扫描 Sub2(见表 3),匹配结果为

表 2 属性列表 props

属性标号	属性值对	比较结果
1	Publisher = "QReilly"	F
2	Author = "Jack"	T

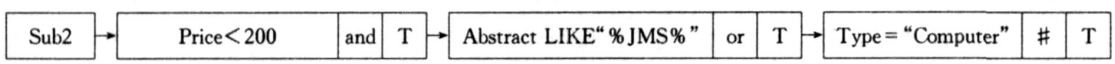


表 3 属性列表 props

属性标号	属性值对	比较结果
1	Publisher = "QReilly"	F
2	Author = "Jack"	T
3	Price < 200	T
4	Abstract LIKE"%JMS%"	T

匹配 Sub2 的过程类同于 Sub1 的过程,最后计算 Sub2 的匹配结果为 T,把 Sub2 添加到匹配列表中,继续 Sub3 的匹配(见表 4),结果为

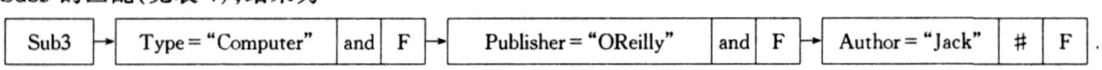
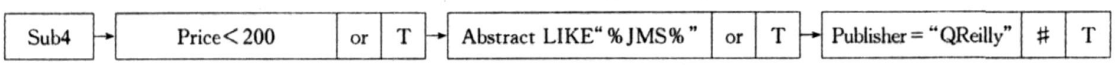


表 4 属性列表 props

属性标号	属性值对	比较结果
1	Publisher = "QReilly"	F
2	Author = "Jack"	T
3	Price < 200	T
4	Abstract LIKE"%JMS%"	T
5	Type = "Computer"	F

最后计算 Sub3 的匹配结果为 F,继续 Sub4 的匹配,匹配结果为



扫描过 Sub4 以后,属性列表和上面相同.它的匹配结果为 T,把该订阅添加到匹配列表,到此匹配完成.

3 算法评估

对提出的算法进行了模拟实验,以验证算法的正确性并对其性能进行评价.实验系统的编程语言为 C#,所使用的机器为一台 CPU 为 Intel Pentium Ⅳ2.4GHz,内存为 512 Mb 的普通台式电脑,所使用的操作系统为 Windows 2003 Server,C# 运行环境为 NET 2005,SDK 2.0.在实验中通过建立 50 个消息发布者持续地向服务器发送消息,在客户端建立一定的不同数量的订阅者从服务器接收消息,通过改变订阅事件属性的个数和订阅数量的不同来比较 3 种算法的优劣.结果如表 5 和表 6 所示.

表 5 事件属性个数渐增时,客户每秒收到的匹配成功订阅量

事件属性个数	记数法	匹配树算法	新的算法
5	43	80	150
10	28	56	126
15	12	33	118
20	5	24	90

表 6 订阅数量渐增时,客户每秒收到的匹配成功订阅量

订阅数量	记数法	匹配树算法	新的算法
500	200	240	280
1 000	120	140	200
1 500	40	80	160
2 000	19	42	130

从上面的实验结果可以发现,事件的属性个数越多,订阅者的数量越多时,需要匹配的属性就越多,这样需要消耗的匹配代价就越高,3 种算法呈现出相同的趋势即客户每秒收到的匹配成功的订阅量就越少.当订阅数量少,订阅条件的属性个数比较少的时候,3 种算法的效率差别不是很大,但是随着订阅数量增加时,所提出的算法明显地表现出高效性,特别是当事件的属性个数增加时,笔者提出的算法的效率要比记数法高几十倍,同样是衰减的趋势,但是该算法相比较其他算法而言效率衰减幅度是最少最优的,这说明其具有良好的扩展性,适合应用在大规模系统中.

4 小 结

近年来基于内容的发布/订阅模型逐渐兴起并得到广泛的应用.如何高效地实现消息与订阅的匹配是该模型实现中的一个关键问题.分析了当前已有的几种基于内容的发布/订阅系统匹配算法的优缺点,提出了一种新的匹配算法,它利用了逻辑操作符所具有的特性,实现了高效快速地匹配.最后实验证明了该算法的正确性和高效性,其效率确实比其他常用算法有很大提高,并且扩展性好,适合应用于大规模分布式基于内容的发布订阅系统中.下一步的工作是将该算法集成到基于内容的发布订阅原型系统中,并进一步研究基于内容的路由算法.

参考文献:

[1] 薛涛,冯博琴,李波,等.基于内容的发布订阅系统中快速匹配算法研究[J].小型微型计算机系统,2006,(3):529-533.

[2] 马建刚,黄涛,汪锦岭,等.面向大规模分布式计算发布订阅系统的核心技术[J].软件学报,2006,(1):137-147.

[3] LIU Hai-feng,MILENKO P,JACOBSEN H A. Efficient and Scalable Filtering of Graph-based Metadata[J]. Web Semantics: Science,Services and Agents on the World Wide Web,2005,(9):294-310.

[4] 姚刚,邓江沙.基于JMS的消息过滤算法[J].计算机技术与发展,2006,(7):91-93.

Efficient Matching Algorithm in Content-based Publish/subscribe Model

ZHANG Cai-yun, KANG Ya-nan, CHENG Ru-zhen

(College of Mathematics and Information Science,Hebei Normal University,Hebei Shijiazhuang 050016,China)

Abstract: A key issue when designing and implementing large-scale content-based publish/subscribe systems is how to efficiently match high volumes of events against large numbers of subscription. A fast and efficient algorithm is presented,according to the characteristic of logical expression,this algorithm makes pretreatment with all the subscription by the priority,in this way,the comparison of the same attributes is one time or less than one time,there by reducing the cost of matching. Experiment proves that it is more efficient than other commonly used algorithms,and has good expansibility,suits large-scale distributed content-based publish/subscribe system.

Key words: content-based; publish/subscribe; matching algorithm; priority

(责任编辑 白占立)