

基于内容的发布订阅系统的一种快速匹配算法

陈 娱,刘健波

(四川大学 计算机学院,四川 成都 610065)

摘要: 目前基于内容的发布/订阅系统得到了广泛的应用,而事件和订阅的匹配算法是其中的一个关键问题。提出了一种高效的匹配算法,首先根据谓词类型和名称的不同建立若干订阅树,建立一个索引结构管理这些树。匹配时,根据事件的类型和名称在对应的树中进行搜索。实验证明该算法具有较好的匹配性能。

关键词: 发布/订阅;匹配算法;多维索引

中图分类号: TP393

文献标识码: A

文章编号: 1674-7720(2012)02-0041-03

Effective matching algorithm in content-based publish/subscribe system

Chen Yu, Liu Jianbo

(Computer College of Sichuan University, Chengdu, 610065, China)

Abstract: Matching algorithm is one of the key problems in content-based publish/subscribe system. This paper proposed an efficient matching method between subscriptions and events. Firstly, construct a tree for each different types of predicate, and manage these trees with a multi-dimension index structure. Secondly, search in the corresponding tree to match an event by its type and name. The experiment result proves that this algorithm works with a high efficiency.

Key words: publish/subscribe; match algorithm; multi-dimension index

发布/订阅系统是一种用于信息交互的中间件系统,它将信息的发布者与订阅者联系在一起。发布者只负责发布信息给中间件,订阅者也只向中间件订阅自己感兴趣的信息,信息具体的发布和传送则由发布/订阅系统负责,这样大大降低了系统的耦合性,使通信更加灵活,符合分布式系统的需求,所以在分布式系统中得到了广泛的应用。

在基于内容的发布/订阅系统中,信息由事件来表示,订阅者可以通过订阅事件来获取信息。一个事件可以表示为一些属性的集合,而订阅则可以表示为一些谓词的集合。订阅者可以通过指定其感兴趣的谓词来灵活地订阅事件。相对于基于主题的发布/订阅系统,基于内容的发布/订阅系统中订阅的表达得到了很大的提高,但是同时系统中的订阅数目也大大增加,匹配的复杂度大大提高,必须有一个高效的算法来实现订阅和事件的快速匹配^[1]。匹配算法的基本思想是尽量优化订阅结构,减少匹配时订阅条件中重复部分的判断,提高匹配效率。

1 相关研究

目前相关的研究已经提出了很多比较有代表性的

算法^[2-6]。Aguilera 等提出了基于搜索树的算法^[4],这种方法建立一棵搜索树,每个非叶子节点表示一个对订阅条件的测试,每条边代表测试结果,每个叶子节点表示一个订阅条件。当匹配一个事件时,如果按照树的某一路径可以从树根到达一个叶子节点,说明该叶子节点代表的订阅与这个事件相匹配。这种方法中相同的订阅条件只需要测试一次,但是节点间的耦合性较高,相应地,维护树的代价也比较高。

计数法的思想是测试所有谓词,建立一个表保存谓词和订阅的映射关系,即谓词被哪些订阅满足。匹配时,遍历这个表找出满足一个订阅的谓词数目,将之与这个订阅包含的谓词数目进行比较,如果相等,则说明事件与订阅相匹配。计数法虽然避免了一个谓词被多次测试,但是它总会对订阅中所有的谓词进行测试,而实际上当一个谓词无法匹配时,其他的测试都不需要继续进行。

2 匹配算法

本文提出的方法借鉴了基于搜索树的方法,学习了其中的优点,并针对其中存在的缺点进行了改进。搜索树的方法将所有谓词添加到一棵树中,树的深度和耦合

网络与通信 Network and Communication

度比较大,不利于订阅树的维护。所以本文方法中修改了构造订阅树的方法,按照谓词类型和名称的不同创建若干订阅树,将不同的谓词放到不同的树中,减少了耦合度。为了便于管理这些搜索树,建立了一个索引结构,根据谓词类型和名称的不同可以快速找到对应的订阅树。

2.1 订阅模型和事件模型

在基于内容的发布/订阅系统中,为了能让订阅者快速、准确地指定所需信息,订阅者要通过一定的订阅模型来订阅信息,发布者也要通过一定的事件模型来发布数据,所以在提出匹配算法之前,首先定义如下的订阅模型和事件模型:

(1) 事件

事件包含了一些发布者发布的数据,可以定义为一些属性的集合。属性是最小的数据,可以表示为一个数据类型、属性名称和属性值组成的三元组,即 $Attribute := \langle type, name, value \rangle$ 。属性的数据类型定义有数值类型和字符串型。事件可以表示为 $\{A_1, A_2, \dots, A_n\}$, 即 $Event := \cup Attribute$ 。

例如事件 $E1 = \{(int, x, 10), (string, s, "teacher")\}$ 包含两个属性 $A_1 = (int, x, 10)$ 和 $A_2 = (string, s, "teacher")$ 。

(2) 订阅

一个订阅表示了一个订阅者所感兴趣的数据,可以定义为一些谓词的集合。谓词是对一个属性的测试结果,可以表示为一个数据类型、属性名称、测试操作符和匹配值组成的四元组,即 $Predicate := (type, name, operator, value)$ 。谓词的测试操作符(operator)对应的数值型定义有 $=, >, <, >=, <=$ 和 $!=$, 对应字符串型定义有 $=, !=$ 和 $*=$ (包含)。订阅可以表示为 $\{P_1, P_2, \dots, P_n\}$, 即 $Subscription := \cup Predicate$ 。

例如订阅 $S1 = \{(int, x, >, 10), (string, s, !=, "student")\}$, 包含两个谓词 $P_1 = (int, x, >, 10)$ 和 $P_2 = (string, s, !=, "student")$ 。

定义 1. 属性 $a(type_a, name_a, value_a)$ 与谓词 $p(type_p, name_p, operator_p, value_p)$ 匹配, 则要满足 $(type_a = type_p) \wedge (name_a = name_p) \wedge (operator_p(value_a, value_p) = true)$ 。

定义 2. 事件 e 与订阅 s 匹配, 则对于 s 中的每个谓词 p , e 中至少存在一个属性 a 与 p 匹配。

定义 3. 谓词 $p1$ 与谓词 $p2$ 间存在覆盖关系, 则对任意一个属性 a , 如果它和 $p2$ 匹配, 则它也一定和 $p1$ 匹配。

定义 4. 谓词 $p1$ 与谓词 $p2$ 冲突, 则当 $p1$ 成立时, $p2$ 一定不成立。

2.2 订阅树的构造

在构造订阅树之前,增加一个对订阅集合预处理的步骤。这是因为在一个订阅中可能存在着冲突和重复的谓词,在订阅和匹配开始前就进行处理可以付出最小的代价。进行预处理时要遍历所有订阅,对于谓词存在冲突的订阅,由于一定不存在事件能与之匹配,所以

不需要将之添加到订阅树中;对于谓词间存在覆盖关系的订阅,只保留最大的谓词,这样可以减少订阅中谓词的重复。

接下来根据上一步经过预处理的订阅集合来创建若干订阅树,将类型和名称相同的谓词生成的节点添加到一棵订阅树中。订阅树中的每个节点存储了一个谓词和包含这个谓词的订阅的集合。为了便于在订阅树中进行匹配,分别以谓词的类型、名称和操作符为键建立一个多级索引结构来管理所有的订阅树,如图 1 所示,将指向每棵订阅树根节点的指针对应存储在索引结构中,这样当需要匹配一个事件的属性时就能根据属性的类型和名称进行快速定位。

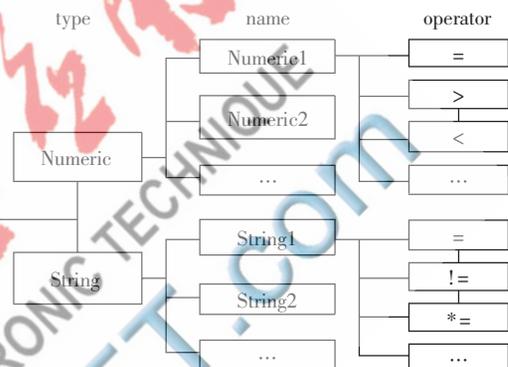


图 1 谓词的索引结构

构建订阅树时要考虑到谓词间的覆盖关系,让父节点的谓词覆盖子节点的谓词,这样进行事件匹配时,如果父节点的谓词与事件的对应属性不匹配,则属性与子节点的谓词也一定不匹配,加快了匹配的速度。

在订阅树中增加节点时,先通过索引结构找到对应的树,然后在树中查找这样一个节点:这个节点覆盖了新增加的节点,而它的子节点都不覆盖新节点,然后将新节点插入到这个节点和它的子节点之间,新节点的订阅集合也要添加相应的订阅者。

在订阅树中删除节点时,先通过索引结构找到对应的树,从根节点遍历树,如果找到一个节点被要删除的节点覆盖,则对对应的订阅者从以这个节点为根的树中的所有节点的订阅集合中删除,如果删除后某个节点的订阅集合为空,则删除该节点。

下面给出了构建订阅树的算法伪代码:

```

ADD-SUBSCRIPTION (sub)
  for each predicate  $P_i$  in sub
    do find root node of the corresponding tree by the
       type and name of  $P_i$ 
       if the tree doesn't exist
         create a new tree
         ADD-PREDICATE (rooti,  $P_i$ )
       ADD-PREDICATE (node, p)
       if the predicate of node covers p
         for each child of node childi

```

网络与通信 Network and Communication

```
do if the predicate of childi covers p
then ADD-PREDICATE (childi, p)
else if p covers the predicate of childi
then insert p between node and childi
else save p as a child of node
```

2.3 事件匹配

由于订阅树的构造方法的改变,事件匹配的方法也有所不同,采用了先找出所有不匹配的订阅,然后得到匹配的订阅方法。匹配一个事件时,先对所有谓词进行测试,对事件中的每个属性,依次查找类型和名称对应的订阅树。如果找到了相应的订阅树,则用这个属性测试订阅树中所有的节点。在测试的过程中,如果一个节点测试失败,则记录下这个节点的订阅者和以这个节点为根的子树中所有节点的订阅者。测试完成后,在订阅集合中剔除所有不匹配的订阅,就能找出所有与当前事件匹配的订阅。

3 实验分析

实验采用的计算机采用双核 Core2、主频为 1.6 GHz 的 CPU, 2 GB 的 DDR2 内存, Windows XP 系统, 算法的实现采用 Visual Studio 2008 平台, C++ 语言。编写了一个订阅产生器和事件产生器来产生实验样本, 每个实验结果为 10 次运行的平均值。每个订阅的形式如 (numeric1>10, numeric2 != 8, string1 *= "student"), 每个事件的形式如 (int1=12, int3=200, string2="teacher")。

实验一 测试订阅数量对匹配速度的影响。首先向系统中添加订阅, 订阅数目变化范围为 1 000~10 000 个, 每个订阅包含 5~10 个谓词; 然后随机生成一个事件, 这个事件包含的属性个数为 20, 测试匹配这个事件所需的时间。实验结果如图 2 所示。

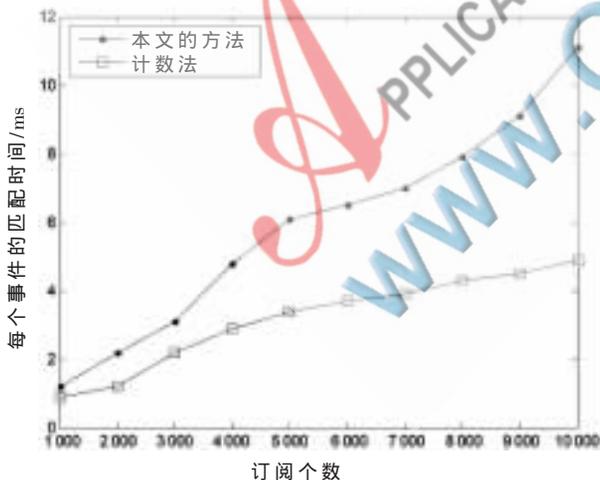


图 2 订阅个数对算法的影响

实验二 测试事件属性个数对匹配速度的影响。向系统中添加 5 000 个订阅, 每个订阅包含 5~10 个谓词, 随机生成事件, 事件的属性个数变化范围为 5~20 个, 测试匹配事件所需的时间。实验结果如图 3 所示。

从上面的实验结果可以看出, 本文提出的算法在匹

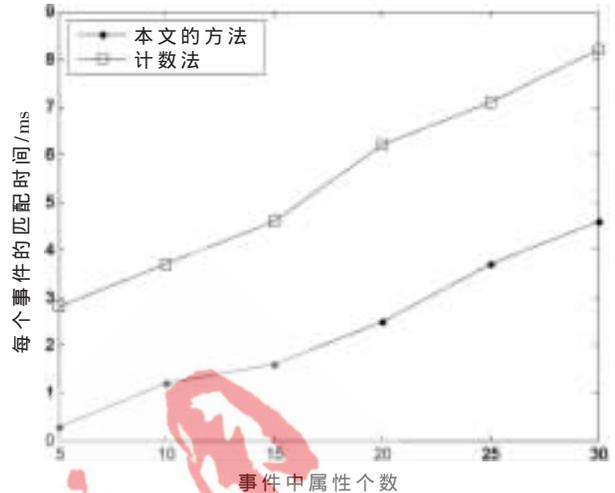


图 3 事件属性个数对算法的影响

配效率上相对于计数法和基于匹配树的算法有了较大的提高。当事件的属性个数相同而系统订阅数不断增加时, 和系统中订阅数目相同而事件包含的属性个数不断增长时, 这种算法明显表现出了它的高效性。

近年来基于内容的发布/订阅系统逐渐获得了越来越广泛的应用。本文针对基于内容的发布/订阅系统中的事件与订阅的匹配, 提出了一种匹配算法。这种算法利用一个多级索引结构来管理不同类型和名称谓词, 提高了谓词的匹配速度, 而且充分考虑了谓词间的覆盖关系, 减少了匹配次数。最后通过实验证明了这种算法的正确性和高效性。未来可以对算法的覆盖关系、订阅树的结构等进行更进一步的改进。

参考文献

- [1] 马建刚, 黄涛. 面向大规模分布式计算发布订阅系统核心技术[J]. 软件学报, 2006, 17(1): 134-147.
- [2] KALE S, HAZAN E, CAO F, et al. Analysis and algorithms for content-based event matching[C]. In proceedings of ICDCS Workshops, 2005; 363-369.
- [3] FABRET F, LLIRBAT F, PEREIRA J, et al. Efficient matching for content-based publish/subscribe systems[C]. In Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles, 2003, 37(5): 29-43.
- [4] AGUILERA M K, STORM R E, STURMAN D C, et al. Matching events in a content-based subscription system[C]. In proceedings of the 18th ACM Symposium on Principles of Distributed Computing, 1999; 53-61.
- [5] 齐凤亮, 金蓓弘. 发布/订阅系统中的原子订阅管理和匹配[J]. 软件学报, 2009, 36(12): 111-114.
- [6] 薛涛, 冯博琴. 基于内容的发布订阅系统中快速匹配算法的研究[J]. 小型微型计算机系统, 2006, 27(3): 529-535.

(收稿日期: 2011-09-22)

作者简介:

陈娱, 女, 1987 年生, 硕士, 主要研究方向: 图形图像。