计算机研究与发展
Journal of Computer Research and Development
ISSN 1000-1239/CN 11-1777/TP
47(10): 1690-1699, 2010

发布/订购系统中基于重复属性判定的事件匹配算法研究

刘 国 周 忠 吴 威
(虚拟现实技术与系统国家重点实验室(北京航空航天大学) 北京 100191)
(北京航空航天大学计算机学院 北京 100191)
(liuguoster@gmail.com)

Event Matching Algorithm Based on the Judgment of Redundant Attributes in Publish/Subscribe Systems

Liu Guo, Zhou Zhong, and Wu Wei
(*State Key Laboratory of Virtual Reality Technology and Systems (Beihang University), Beijing* 100191)
(*School of Computer Science and Engineering, Beihang University, Beijing* 100191)

Abstract In the Map based publish/subscribe systems, most of the typical event matching approaches start from the published events, and then move through to looking for the matched subscriptions. Since there are always some redundant attributes in different events, the same attribute would probably exist in more than one event. When the number of published events is big, the same attributes in different events would be matched more than once with the constraints in the subscriptions, i.e., there are redundant matches during the event matching process. To alleviate this redundant matching problem, a new event matching algorithm which is based on the judgment of redundant attributes in different events is presented in this paper. By judging the redundancy of attributes, merging the event sets to eliminate the matching redundancy, and maintaining the constraints in subscriptions set into a multi-level index structure, this new event matching algorithm improves the matching efficiency in time complexity and also the maintainability in space complexity. Based on the above approach, a series of experiments with comparison are made. The experiment results clearly show that this event matching algorithm has higher efficiency compared with similar approaches when the number of events and subscriptions is big approaching hundreds or thousands.

Key words distributed system; publish/subscribe; event/subscription; attribute/constraint; event matching algorithm; redundant match

摘 要 在基于Map的发布/订购系统中,典型的事件匹配算法大都针对用户发布的每个事件寻找相匹配的订购,由于同一属性在不同事件中重复出现是一种普遍现象,当用户发布的事件数量较大时,相同的属性会与订购中的约束条件重复匹配,事件匹配存在着冗余.针对这种重复匹配问题,提出一种基于重复属性判定的事件匹配算法,该算法通过判定属性的重复关系,合并事件集合去除重复属性,并将订购集合组织为约束的多级索引结构以减少不必要的匹配,从而提高算法的匹配效率和可维护性.实验表明,当事件数量和订购数量较大时,该算法与同类算法相比具有更高的匹配效率.

关键词 分布式系统;发布/订购;事件/订购;属性/约束;事件匹配算法;重复匹配

中图法分类号 TP393

收稿日期:2008-01-18;**修回日期**:2010-04-26

基金项目:国家"九七三"重点基础研究发展计划基金项目(2009CB32085);2008年下一代互联网应用示范项目子课题基金项目(CNGI2008-123);中央高校基本科研业务费专项基金项目

0 引 言

发布/订购模型是分布式环境中应用程序间相互通信的一种通信模型,具有异步、松散耦合和多对多通信的特点,适应动态多变的分布式系统的需求,得到了广泛的研究[1],基于此模型的发布/订购系统使分布式系统中的各参与者之间能以发布/订购的方式进行交互.从订购语言的角度可以将发布/订购系统分为基于主题的和基于内容的,基于内容的发布/订购系统允许订购者在事件的内容上指定约束条件,具有订购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购中的每个约束条件进行匹配,由于基于内容的发布/订购系统中组成每个订购的约束条件数目往往较大,因此对事件匹配的效率要求较高.发布/订购系统中的事件匹配算法负责找到与给定的事件相匹配的所有订购,由于事件匹配算法在基于内容的发布/订购系统中具有重要的作用,因此得到了较多的研究.本文研究基于内容的发布/订购系统中的事件匹配算法.

1 相关工作

发布/订购系统中的事件匹配算法负责找到与给定的事件相匹配的所有订购.衡量一个事件匹配算法优劣的标准主要有:匹配的时间效率、匹配的空间效率和事件/订购可维护的效率,其中匹配的时间效率是最重要的标准.基于内容的发布/订购系统可以分为两类[2-3]:一类是基于 Map 的,另一类是基于 XML 的.在基于 Map 的发布/订购系统中,事件的内容为多个"属性=值"的集合,每个集合称之为一个 Map,典型的事件匹配算法主要有:基于搜索树的算法[4]、基于并行搜索树的算法[5]、基于二叉判定图的算法[6]、文档计数算法[7]及其改进算法[8-9].在基于 XML 的发布/订购系统中,每个事件是一个 XML 文档,用户的订购条件一般是 XPath 表达式,典型的事件匹配算法主要有:XFilter 方法[10]、YFilter 方法[11]、基于 XTrie 索引结构的方法[12]和基于层级下推自动机的方法[13].

基于 Map 的发布/订购系统中现有的事件匹配算法大都是将用户的订购条件组织为特定的数据结构,对组织好的数据结构进行遍历,从而得到与事件相匹配的订购条件.基于搜索树的算法将订购条件组织为一种树型结构,系统按照某一路径从根节点搜索到某叶子节点,就找到了与事件相匹配的所有订购条件,该算法的时间复杂度较低.基于并行搜索树的算法用树中每个叶子节点表示一个订购条件,每个非叶子节点表示对某属性的一个操作,每条边表示操作的结果,系统通过对该搜索树进行遍历得到与事件相匹配的所有订购条件,该算法对订购条件中相同的约束只作一次判断,时间效率较好.基于二叉判定图的算法将订购条件中的每个约束表示为一个布尔条件,将订购条件表示为一个二叉判定图,匹配过程中先求出订购条件中各约束对应的布尔变量值,然后遍历该二叉判定图得到与事件相匹配的所有订购条件,该算法的优点在于它可以同时支持"与"和"或"两种操作.文档计数算法及其相关改进算法则是比较订购包含的约束个数和事件所能匹配的约束个数是否相等来判断该事件和订购是否匹配.在以上算法中,基于搜索树的算法将所有订购组织为一种树形结构,空间复杂度和维护的成本很高;基于并行搜索树的算法将订购集合组织为一个并行搜索树,但是不同的约束往往具有不同的类型,所支持的操作符也不同,而事件匹配过程需要遍历整个搜索树,因此会造成大量无意义的操作;基于二叉判定图的算法需要两次遍历整个二叉判定图,时间复杂度较高.相比之下,文档计数法通过判定约束的重复关系减少匹配次数,文档计数法改进算法在此基础上,将相同类型的约束组织在同一链表或 Hash 表中,对同一链表中的所有约束,再利用约束的包含关系进行排序,这样,当某一事件发布时,通过约束的多级索引结构即可快速定位该事件所对应的约束链表或 Hash 表,然后通过二分查找法或计算 Hash 值的方法找到与之匹配的所有约束条件.该算法利用约束的重复关系和包含关系减少不必要的匹配,并将订购集合组织为约束的多级索引机构以避免大量无意义的操作,因而具有更高的匹配效率.

在基于 Map 的发布/订购系统中,这些事件匹配算法都针对用户发布的每个事件寻找相匹配的订购条件,当用户发布的事件数量较大时,同一属性在不同事件中重复出现的现象较为普遍,由于这些算法均未考虑属性的重复问题,相同的属性会被重复匹配,事件匹配存在着冗余,影响了算法的匹配效率.

针对这种相同属性的重复匹配问题,本文面向基于 Map 的发布/订购系统开展研究,提出一种基于重复属性判定的事件匹配算法.和以上算法不同,本算法引入了对事件重复属性进行判定的思想,通过去除事件的重复属性,将相同的属性只与约束条件进行一次匹配操作,并结合文档计数算法及其索引结构思想[7]将订购集合组织为多级索引结构来提高匹配效率.最后本文进行了实验验证.

2 重复属性判定的思想与定义

我们提出的基本思想:对于用户发布的事件集合,通过判定事件集合中属性的重复关系而使相同的属性只与订购条件进行一次匹配操作;对于用户的订购集合,通过判定订购集合中约束的重复关系而使事件与相同的约束只进行一次匹配操作,通过判定订购集合中约束的包含关系而利用约束间的包含关系减少与事件的匹配次数.主要包括以下3种情况:

1. 设有事件 E_p,E_q 和 E_r,分别包含了属性 a_p,a_q 和 a_r,如图1所示,若 a_p,a_q 和 a_r 是相同的属性,则当用事件 E_p,E_q,E_r 匹配订购 S 时,只需判断 a_p,a_q,a_r 其中一个是否与 c 匹配即可;

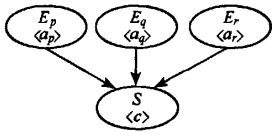

Fig. 1 Event matches subscription (many to one).
图1 事件与订购匹配示意图(多对一)

2. 设有订购 S_p,S_q 和 S_r,分别包含了约束 c_p,c_q 和 c_r,如图2所示,若 c_p,c_q 和 c_r 是相同的约束,则当用事件 E 分别匹配订购 S_p,S_q,S_r 时,只需判断 a 与 c_p,c_q 和 c_r 其中一个是否匹配即可;

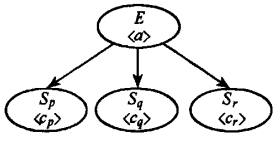

Fig. 2 Event matches subscription (one to many).
图2 事件与订购匹配示意图(一对多)

3. 设有订购 S_p,S_q 和 S_r,分别包含了约束 c_p,c_q 和 c_r,如图2所示,若 c_p,c_q,c_r 间存在包含关系使得对任意属性 a,只要 a 与 c_p 匹配,有 a 与 c_q,c_r 也一定匹配,则当用事件 E 分别匹配订购 S_p,S_q,S_r 时,若能够判定 a 与 c_p 匹配,则根据 c_p,c_q,c_r 间的包含关系即可得到 a 与 c_q,c_r 匹配.

为了具体阐述本论文的事件匹配算法和匹配过程,我们给出相关定义如下:

定义 1. 属性.属性是组成事件的元素,属性定义为包括类型、名称和值的元组,即 $attribute=\langle type, name, value\rangle$.其中,$type$ 表示该属性的数据类型,基于系统需求的不同,预定义的类型集可以包含不同的数据类型;$name$ 表示该属性的名称,定义为字符串;$value$ 即为该属性的取值,它的值域就是该属性的数据类型所能表示的范围,根据数据类型和系统需求不同而不同.

定义 2. 约束.约束是一个原子命题,它是一个无状态的布尔表达式,约束定义为包括类型、名称、操作和值的元组,即 $constraint=\langle type, name, operator, value\rangle$.其中,$type$ 表示该约束所针对属性的数据类型;$name$ 表示该约束所针对属性的名称,用字符串描述;$operator$ 为该数据类型所对应的操作符,根据具体数据类型和系统需求的不同而有所不同;$value$ 是具体的匹配阈值,包含在该属性的值域范围之内.

定义 3. 事件.事件定义为属性的集合,即 $Event=\bigcup attribute$.

定义 4. 订购.订购定义为约束的集合,即 $Subscription=\bigcup constraint$.

定义 5. 属性间的相等关系.对于属性 $a_1=\langle t_{a_1}, n_{a_1}, v_{a_1}\rangle$ 和 $a_2=\langle t_{a_2}, n_{a_2}, v_{a_2}\rangle$,若满足 $P(a_1,a_2)=(t_{a_1}=t_{a_2})\wedge(n_{a_1}=n_{a_2})\wedge(v_{a_1}=v_{a_2})$ 为真,则称 a_1 等于 a_2,记为 $a_1=a_2$.即 $\forall a_1\,\forall a_2, P(a_1,a_2)\rightarrow a_1=a_2$.

定义 6. 属性和约束的匹配关系.若属性 $a=\langle t_a, n_a, v_a\rangle$ 与约束 $c=\langle t_c, n_c, op_c, v_c\rangle$ 之间满足 $Q(a,c)=(t_a=t_c)\wedge(n_a=n_c)\wedge(op_c(v_a,v_c)=\text{true})$ 为真,则称属性 a 匹配约束 c,记作 $a\propto c$.即 $\forall a\,\forall c, Q(a,c)\rightarrow a\propto c$.

定义 7. 约束的包含关系.对于约束 c_1,c_2 和任意属性 a,若属性 a 匹配约束 c_1,有 a 也一定匹配约束 c_2,则称 c_1 包含于 c_2 或 c_2 包含 c_1,记作 $c_1\prec c_2$.即 $\forall a, (a\propto c_1\rightarrow a\propto c_2)\rightarrow c_1\prec c_2$.

定义 8. 约束的相等关系.对于约束 c_1 和 c_2,若 c_1 包含 c_2 并且 c_2 也包含 c_1,则称 c_1 等于 c_2,记作 $c_1=c_2$.即 $\forall c_1\,\forall c_2, (c_1\prec c_2\wedge c_2\prec c_1)\rightarrow c_1=c_2$.

定义 9. 事件和订购的匹配关系.对于订购 S 中的任何约束 c,若事件 E 中至少存在一个属性 a,使得 $a\propto c$,则称事件 E 匹配订购 S,记为 $E\propto S$.即 $\forall c\in S, \exists a\in E, a\propto c\rightarrow E\propto S$.

在发布/订购系统中,用户对于数据的更新以事件的形式发布,每一个事件是一系列属性的集合,而用户要访问什么样的数据,则以订购条件的形式描述,每个订购是一系列约束的集合,定义1~4分别给出了它们的定义.其中,属性和约束是发布/订购系统最基本的元素,它们分别组成了用户发布的事件和订购的条件,根据系统需求的不同,组成属性和约束的类型、名称、操作符和取值也有所不同.在此,我们取属性和约束的类型 $type$ 共有4种,分别为 Integer, Float, String 和 Bool,所支持的操作符有

$>$,$<$,$\geqslant$,$\leqslant$,$=$,前缀 prefix、后缀 suffix 和子串 substring,其中后 3 个操作符是仅针对 String 类型的,这些类型和操作符能够满足大部分发布/订购系统的需求.

在基于重复属性判定的事件匹配算法中,为了便于判定属性间和约束间的相关关系,给出相关定理如下:

引理 1. 设 op_1,op_2 为操作符,函数 $f(op_1,op_2)$ 是以 op_1,op_2 为自变量的函数,取值如表 1 所示,则有

$$\forall v\forall v_1\,\forall v_2,((op_1(v,v_1)=\text{true})\wedge(op(v_1,v_2)=\text{true}))\rightarrow op_2(v,v_2).$$

定理 1. 约束的包含关系. 对于约束 $c_1=\langle t_{c_1},n_{c_1},op_{c_1},v_{c_1}\rangle$ 和 $c_2=\langle t_{c_2},n_{c_2},op_{c_2},v_{c_2}\rangle$,若满足 $(t_{c_1}=t_{c_2})\wedge(n_{c_1}=n_{c_2})\wedge(op(v_{c_1},v_{c_2})=\text{true})$ 为真,则有 $c_1\prec c_2$ 成立,其中 $op=f(op_{c_1},op_{c_2})$,函数取值如表 1 所示. 即

$$\forall c_1\,\forall c_2,((tc_1=tc_2)\wedge(nc_1=nc_2)\wedge(op(vc_1,vc_2)=\text{true}))\rightarrow c_1\prec c_2.$$

Table 1 Function $f(op_1,op_2)$ Value Table

表 1 函数 $f(op_1,op_2)$ 取值表

op_1	op_2	$f(op_1,op_2)$	op_1	op_2	$f(op_1,op_2)$
$>$	$>$	$\geqslant$	$>$	$\geqslant$	$\geqslant$
$\geqslant$	$>$	$>$	$\geqslant$	$\geqslant$	$\geqslant$
$<$	$<$	$\leqslant$	$<$	$\leqslant$	$\leqslant$
$\leqslant$	$<$	$<$	$\leqslant$	$\leqslant$	$\leqslant$
$=$	$=$	$=$	prefix	prefix	prefix
suffix	suffix	suffix	substring	substring	substring

证明. 已知 $c_1=\langle t_{c_1},n_{c_1},op_{c_1},v_{c_1}\rangle$,$c_2=\langle t_{c_2},n_{c_2},op_{c_2},v_{c_2}\rangle$,$op=f(op_{c_1},op_{c_2})$,满足 $(t_{c_1}=t_{c_2})\wedge(n_{c_1}=n_{c_2})\wedge(op(v_{c_1},v_{c_2})=\text{true})$ 为真,那么:

设 $Q_1(a,c_1)=(t_a=t_{c_1})\wedge(n_a=n_{c_1})\wedge(op_{c_1}(v_a,v_{c_1})=\text{true})$,$Q_2(a,c_2)=(t_a=t_{c_2})\wedge(n_a=n_{c_2})\wedge(op_{c_2}(v_a,v_{c_2})=\text{true})$,对任意 $a=\langle t_a,n_a,v_a\rangle$,若 $a\infty c_1$ 成立,即 $Q_1(a,c_1)$ 为真,则 $op_{c_1}(v_a,v_{c_1})=\text{true}$ 成立,根据题设知 $op(v_{c_1},v_{c_2})=\text{true}$ 成立,根据引理 1,有 $op_{c_2}(v_a,v_{c_2})=\text{true}$ 成立,结合题设条件,有 $Q_2(a,c_2)$ 为真,即 $a\infty c_2$ 成立. 证毕.

可以证明,定理 1 是判断两个约束是否具有包含关系的充分条件,而非必要条件.

定理 2. 约束的相等关系. 对于约束 $c_1=\langle t_{c_1},n_{c_1},op_{c_1},v_{c_1}\rangle$ 和 $c_2=\langle t_{c_2},n_{c_2},op_{c_2},v_{c_2}\rangle$,若满足 $(t_{c_1}=t_{c_2})\wedge(n_{c_1}=n_{c_2})\wedge(op_{c_1}=op_{c_2})\wedge(v_{c_1}=v_{c_2})$ 为真,则有 $c_1=c_2$ 成立,即

$$\forall c_1\,\forall c_2,((t_{c_1}=t_{c_2})\wedge(n_{c_1}=n_{c_2})\wedge(op_{c_1}=op_{c_2})\wedge(v_{c_1}=v_{c_2}))\rightarrow c_1=c_2.$$

证明. 已知 $c_1=\langle t_{c_1},n_{c_1},op_{c_1},v_{c_1}\rangle$,$c_2=\langle t_{c_2},n_{c_2},op_{c_2},v_{c_2}\rangle$,满足 $(t_{c_1}=t_{c_2})\wedge(n_{c_1}=n_{c_2})\wedge(op_{c_1}=op_{c_2})\wedge(v_{c_1}=v_{c_2})$ 为真,那么:

设 $Q_1(a,c_1)=(t_a=t_{c_1})\wedge(n_a=n_{c_1})\wedge(op_{c_1}(v_a,v_{c_1})=\text{true})$,$Q_2(a,c_2)=(t_a=t_{c_2})\wedge(n_a=n_{c_2})\wedge(op_{c_2}(v_a,v_{c_2})=\text{true})$,对任意 $a=\langle t_a,n_a,v_a\rangle$,若 $a\infty c_1$ 成立,即 $Q_1(a,c_1)$ 为真,则 $op_{c_1}(v_a,v_{c_1})=\text{true}$ 成立,根据题设知 $op_{c_2}(v_a,v_{c_2})=\text{true}$ 成立,结合题设条件,有 $Q_2(a,c_2)$ 为真,即 $a\infty c_2$ 成立;反之,若 $a\infty c_2$ 成立,则有 $a\infty c_1$ 亦成立. 根据定义 8,有 $c_1=c_2$ 成立. 证毕.

在基于重复属性判定的事件匹配算法实现过程中,我们可以用定义 5 判定事件集合所包含的属性是否具有重复关系,而对于订购集合间约束的包含关系和重复关系,由于定义 7 和定义 8 要穷举所有可能的属性,不便于具体实现,因此我们给出了定理 1 和定理 2,分别用于判定约束间的包含关系和重复关系.

3 基于重复属性判定的事件匹配算法

在基于内容的发布/订购系统中,事件与订购之间的匹配过程一般分为两个阶段:用户事件和订购的预处理阶段以及预处理后事件和订购的匹配阶段. 事件和订购预处理阶段处理用户的事件集合和订购集合,并转化为它们的内部数据表示,而匹配阶段将内部表示的订购集合与事件集合相匹配,针对每个事件得到相匹配的订购. 根据事件集合和订购集合预处理方法的不同,产生的内部数据表示和存储的数据结构也不尽相同,从而影响相应的匹配过程和匹配效率,因此事件匹配算法的重要一步就是对事件集合和订购集合的预处理.

在基于内容的发布/订购系统中,当事件数量和订购数量较多时,组成事件的各属性之间可能存在着大量的重复,组成订购的各约束间也可能存在着大量的相等或包含关系. 为了减少不必要的匹配,我们首先将事件集合和订购集合进行预处理,将事件集合合并为属性集合并根据定义 5 去除重复属性,将订购集合合并为约束集合并根据定理 2 去除重复约束,然后根据定理 1 判断约束间的包含关系并组织为多级索引结构,进而完成属性集合和约束集合的匹配过程.

3.1 数据结构

本算法的数据结构主要有:属性-事件链表、约

束-订购链表、订购-事件匹配链表和约束多级索引结构. 在属性-事件链表结构中,每个属性指向一个链表,该链表存储了所有包含此属性的事件,如图 3 所示. 在约束-订购链表结构中,每个约束指向一个链表,该链表存储了所有包含此约束的订购,如图 4 所示. 在订购-事件匹配链表结构中,每个订购指向相匹配的属性所在的事件列表以及该事件所能匹配的约束个数,如图 5 所示,该链表的表头部分在订购集合的预处理过程中初始化,表体部分在属性集合与约束集合的匹配过程中逐步填充,匹配过程结束后,对于该链表结构中的每个订购 S_j,当事件 E_{ij} 所能匹配 S_j 中的约束个数 n_{ij} 等于 S_j 所包含的约束个数 n_j,即 $n_{ij}=n_j$ 时,据定义 9,有 E_{ij} 匹配 S_j.

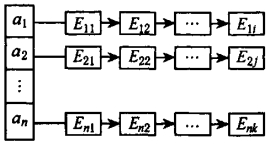

Fig. 3 Attribute-event list.

图 3 属性-事件链表

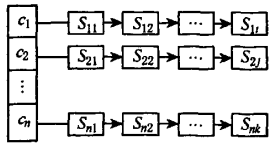

Fig. 4 Constraint-subscription list.

图 4 约束-订购链表

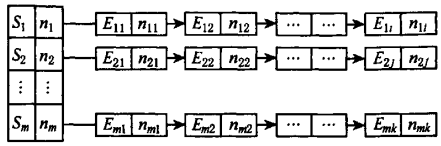

Fig. 5 Subscription-event matching list.

图 5 订购-事件匹配链表

约束的多级索引结构将约束集合按照数据类型、属性名称和操作符逐级建立多级索引,每一个最终索引项是一个约束列表 CL,如图 6 所示. 对于判定大小关系的操作符,CL 中的约束是按照约束的包含关系进行排序的,即有 $\forall c_i, c_j \in CL, i<j \rightarrow c_i \prec$

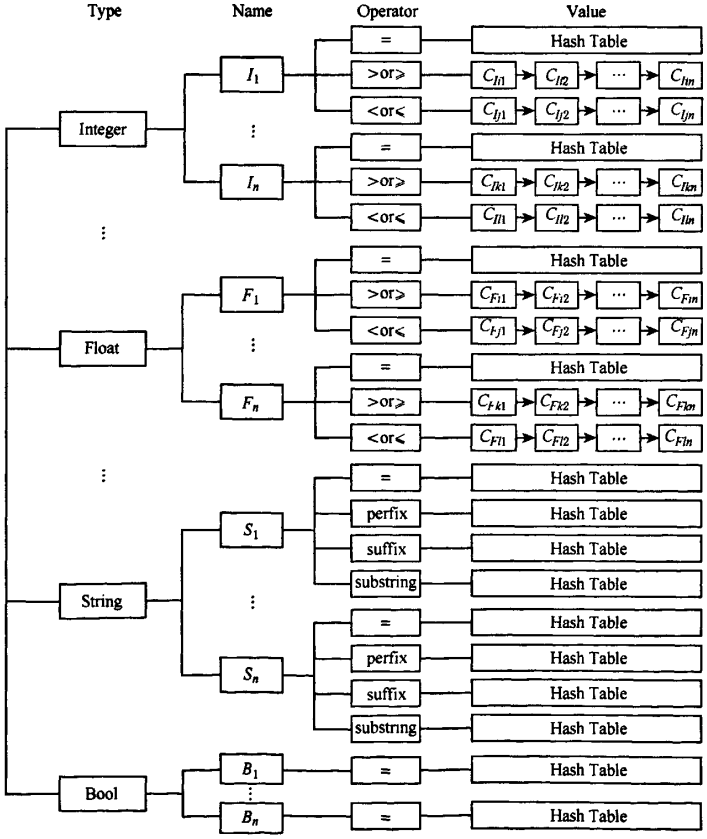

Fig. 6 Constraint multilevel index structure.

图 6 约束的多级索引结构

c_j.给定属性 a,可在对应的 CL 中找到第 1 个与之相匹配的约束 c_i,根据定义 7 可知该 CL 中 c_i 后的所有约束均与属性 a 相匹配,无需逐个进行匹配运算.对于相等操作符以及字符串相关操作符,该 CL 实质是一个 Hash 表,根据约束类型的不同采用的 Hash 函数也不同,给定属性 a,根据该属性的 value 值和相应的 Hash 函数即可快速定位与属性 a 相匹配的约束在 CL 中的位置.

3.2 匹配过程

在本算法中,设事件集合为 E,订购集合为 S,预处理后的属性集合为 A,约束集合为 C,属性-事件链表为 AEList,约束-购链表为 CSList,订购-事件匹配链表为 SEList,约束的多级索引结构为 CIndex.事件和订购的预处理过程接收输入 E 和 S,首先合并事件集合 E 为属性集合 A,根据定义 5 去除重复属性,然后计算订购集合 S 中每个订购包含的约束个数,初始化 SEList 列表,再将集合 S 合并为约束集合 C,根据定理 2 去除重复约束;然后根据属性与事件的从属关系,将集合 A 和 E 组织为 AEList 列表,同样也将集合 C 和 S 组织为 CSList 列表;最后,针对集合 C 中的所有约束,按照定理 1 判定约束间的包含关系,并组织为 CIndex 结构,完成事件集合和订购集合的预处理过程.

算法的匹配过程则首先针对集合 A 中每个属性 a_i 及所支持的操作符 op,查找 a_i 在 CIndex 中相匹配的所有约束,若存在这样的约束 c_j,则根据 a_i 和 c_j 分别从 AEList 和 CSList 中找到对应的事件 E_i 和订购 S_j,并修改 SEList 中相应的节点,将 E_i 所能匹配 S_j 中的约束个数增加 1.由于在约束的多级索引结构 CIndex 中,非 Hash 表的每个约束列表 CL 是按照约束的包含关系排序的,对于在 CIndex 中查找与属性 a_i 向匹配的第 1 个约束 c_j 的过程,我们可以采用二分查找法提高查找效率.匹配过程的伪代码表示如下所示:

```
/*算法匹配过程,匹配结果存储在 seList 链表中*/
PROC MatchProcess(OP opSet,Attribute attrSet,
    Constraint consSet, AEList aeList, CSList
    csList,SEList seList,CIndex cIndex)
  FOR EACH attr IN attrSet
    FOR EACH op IN opSet
      IF(IsValid(attr,op)) THEN
        IF(op="=") THEN
          /*寻找与 attr 相匹配的约束*/
          Constraint cons :=
            GetMatchedCons(attr,op,cIndex);
          IF(cons≠NULL) THEN
            /*存在与 attr 相匹配的约束,修改 seList 中对应的匹配个数项*/
            CALL AddSubsEventMatchCount
              (aeList, attr, csList, cons,
              seList);
          ELSE
            /*寻找约束链表中与 attr 相匹配的第 1 个约束*/
            Constraint c := GetFirstMatchedCons
              (attr,op,cIndex);
            IF(c≠NULL)THEN
              /*对所有与 attr 相匹配的约束,修改 seList 中对应的匹配个数项*/
              FOR EACH cons IN{c,
                constraints BEHIND c}
                CALL AddSubsEventMatchCount
                  (aeList, attr, csList, cons,
                  seList).
END PROC
/*修改订购-事件匹配链表的过程*/
PROC AddSubsEventMatchCount(AEList aeList,
  Attribute attr, CSList csList, Constraint
  cons,SEList seList)
  /*寻找 aeList 中属性 attr 对应的属性-事件链表*/
  AEListNode aeNode := GetAEListNode
    (aeList,attr);
  /*寻找 csList 中约束 cons 对应的约束-订购链表*/
  CSListNode csNode := GetCSListNode
    (csList,cons);
  /*对包含约束 cons 的每个订购*/
  FOR EACH Subscription subs IN csNode
    /*寻找 seList 中订购 subs 对应的订购-事件匹配链表*/
    SEListNode seNode := GetSEListNode
      (seList,subs);
    /*对包含属性 attr 的每个事件*/
    FOR EACH Event evt IN aeNode
```

```
/* 寻找 seNode 中事件 evt 对应的事件-匹配个数节点 */
EventConsCountNode eccNode :=
  GetECNode(seNode,evt);
IF(eccNode=NULL)THEN
  /* 不存在 evt 对应的节点,申请一个事件-匹配个数结点,加入 seNode 链表中 */
  eccNode := NEW EventConsCountNode;
  eccNode.Event := evt;
  eccNode.MatchedConsCount := 1;
  /* 将该节点加入 seNode 链表 */
  CALL seNode.insert(eccNode);
ELSE
  /* 存在 evt 对应的节点,将该节点对应的匹配约束个数增 1 */
  eccNode.MatchedConsCount :=
    eccNode.MatchedConsCount+1.
END PROC
```

这样,我们就可以把订购集合中每个订购所包含的约束个数及其与事件集合的匹配情况记录在订购-事件匹配链表 SEList 中. 对于该链表中的每个订购 S_j,当事件 E_{ij} 所能匹配 S_j 中的约束个数 n_{ij} 等于 S_j 所包含的约束个数 n_j,即 $n_{ij}=n_j$ 时,根据定义9,有 E_{ij} 匹配 S_j. 根据此匹配链表,即可找到那些与用户发布的事件相匹配的所有订购条件,进而通知订购用户相关事件的更新.

4 实验与结果

4.1 实验参数与规则

在基于 Map 的发布/订购系统中,顺序比较法"BruteForce"是最为原始的事件匹配算法,一般在事件匹配算法的改进过程中都会作为参考算法;文档计数法"Counting"通过判定约束的重复关系减少匹配次数,而其改进算法"ICCounting"则通过判定约束的重复关系和包含关系减少匹配次数,并将订购集合组织为约束的多级索引结构以避免大量无意义的操作,较之其他事件匹配算法具有更高的时间效率. 因此我们将基于重复属性判定的事件匹配算法"RABasedMatching"与"BruteForce","Counting"以及"ICCounting"算法进行了对比测试. 在生成事件集合与订购集合并评估算法效率时,我们结合文献[8]中的生成规则与测试参数,给出了本实验中用到的生成规则与测试参数.

实验中所需要的事件集合与订购集合由一个数据产生器生成,其产生规则如下:

令 D_t 表示属性/约束类型总个数,取值为 4,其中 Integer 类型占 40%、Float 类型占 30%、String 类型占 20%、Boolean 类型占 10%;

令 D_n 表示属性/约束名称总个数,取值为 A_n,其中每个名称是字母表中 6 个随机字符组成的字符串;

令 D_{op} 表示约束操作符总个数,为便于性能比较取值为 5,分别是 $>$,$<$,$\geqslant$,$\leqslant$ 和 $=$,比例均为 20%;

令 D_{vi} 和 D_{vf} 分别表示 Integer 和 Float 类型约束值的取值范围,取 $D_{vi}=[-100,100]$,取 $D_{vf}=[-100,100]$,每个 Integer 或 Float 类型约束值为该范围内一个随机值;

令 D_{vs} 表示 String 类型约束值的取值范围,称为 String 型约束取值表,表中每个元素是字母表中 8 个随机字符组成的字符串,取该表长度 $|D_{vs}|=100$,每个 String 类型约束值为该表中一个随机元素;

令 D_{vb} 表示 Boolean 类型约束值取值范围,取 $D_{vb}=\{\text{true},\text{false}\}$,其中 true 和 false 所占比例分别为 50%.

实验中的测试参数及说明如下:令 E 表示事件数量;令 a_l 和 a_h 分别表示每个事件包含属性个数的下限值和上限值,每个事件包含的属性个数取值范围为 $[a_l,a_h]$;令 A_n 表示属性/约束名称表长度;令 S 表示订购数量;令 c_l,和 c_h 分别表示每个订购包含约束个数的下限值和上限值,每个订购包含的约束个数取值范围为 $[c_l,c_h]$. 本实验分别测试了这些参数的变化对匹配效率的影响并给出了测试结果.

4.2 实验结果与分析

本实验是在具有 Pentium(R) 4 CPU 2.80GHz 处理器、512 MB 内存、操作系统为 Microsoft Windows XP Professional 的 PC 机上进行的,实验结果中的每个数据点都是以相同的参数运行 10 次取平均值得到的.

为考察每个测试参数的变化对算法匹配效率的影响,我们针对每个测试参数分别进行了一组实验,在每组实验中,令所要测试的参数分别取不同的值,保持其他参数取固定值,考察参数变化对算法匹配效率的影响. 每组实验的测试参数取值和实验结果如下:

1) $E=1\,000$,$A_n=500$,$S=1\,000$,$c_l=10$,$c_h=15$;$a_l=a_h=20\sim200$,测得结果如图 7 所示:

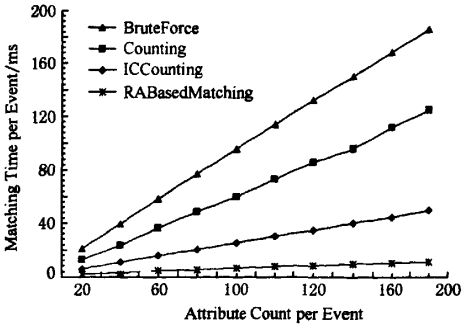

Fig. 7 Attribute count per event's influence.

图 7 事件属性个数的影响

2) $a_l=20$,$a_h=30$,$A_n=500$,$S=1\,000$,$c_l=10$,$c_h=15$;$E=500\sim5\,000$,测得结果如图 8 和图 9 所示:

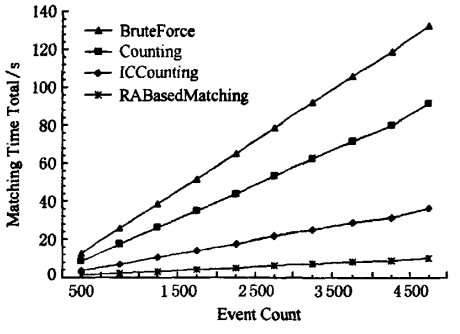

Fig. 8 Event count's influence(1).

图 8 事件数量的影响(1)

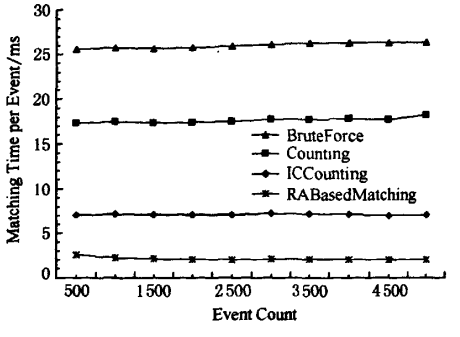

Fig. 9 Event count's influence(2).

图 9 事件数量的影响(2)

3) $E=1\,000$,$a_l=20$,$a_h=30$,$S=1\,000$,$c_l=10$,$c_h=15$;$A_n=50\sim500$,测得结果如图 10 所示:

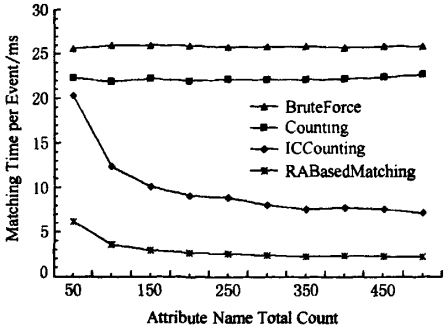

Fig. 10 Attribute name count's influence.

图 10 属性名称表长度的影响

4) $E=1\,000$,$a_l=20$,$a_h=30$,$A_n=500$,$S=1\,000$;$c_l=c_h=5\sim50$ 测得结果如图 11 所示:

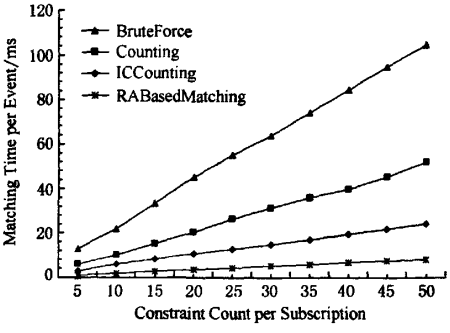

Fig. 11 Constraint count per subscription's influence.

图 11 订购约束个数的影响

5) $E=1\,000$,$a_l=20$,$a_h=30$,$A_n=500$,$c_l=10$,$c_h=15$;$S=500\sim5\,000$,测得结果如图 12 所示:

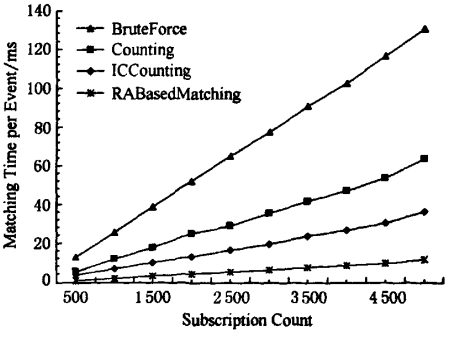

Fig. 12 Subscription count's influence.

图 12 订购个数的影响

由以上实验结果可以看出,当事件数量和订购数量较大时,BruteForce 算法的效率最低,Counting 算法其次,ICCounting 算法由于采用多级索引结构并利用约束间的包含关系减少不必要的匹配,使得

该算法的匹配效率较之 Counting 算法有显著提高，本文所提出的 RABasedMatching 算法由于进一步去除了事件集合中的重复属性，使得匹配效率进一步提高. 根据实验结果，这几种算法的匹配耗时都随着事件数量、事件属性个数、订购数量和订购约束个数的增加而增加，但是 ICCounting 算法和 RABasedMatching 算法的匹配耗时却随着属性名称表长度(A_n)值的增加而降低，A_n的取值对其他两种算法则无明显影响，如图 10 所示，这是因为在约束的多级索引结构下，属性名称表范围越大，属性名称索引就会越分散，从而多级索引的检索速度会越快，匹配效率就会越高. 同时，随着事件数量、事件属性个数、订购数量和订购约束个数的增加，RABasedMatching 算法的匹配效率下降最慢，与同类算法相比效率更高. 由于本算法的数据结构多采用链表结构，因此在系统运行过程中，当事件集合与订购集合动态变化时，这种基于链表结构的事件匹配算法也便于调整数据结构，使得事件集合和订购集合的动态变化不会对算法匹配效率产生明显影响.

5 总 结

本文基于 Map 的发布/订购系统，提出一种基于重复属性判定的事件匹配算法，通过判断事件属性间的重复关系、订购约束间的重复关系和包含关系减少不必要的匹配，从而能够较快地找到与用户发布的事件相匹配的所有订购条件. 实验表明，当事件数量和订购数量较大时，本算法与同类算法相比具有更高的匹配效率.

在分布式发布/订购系统中，每个节点充当一个事件代理，这些代理事件共同构成一个网络. 当一个事件被发布后，如何使该事件沿着事件代理网络中一条恰当的路径，高效、可靠地传送到感兴趣的订购者，是发布/订购系统中消息路由策略所要解决的问题. 在本文的后续研究工作中，将重点研究发布/订购系统中的消息路由策略和路由协议，这对于分布式发布/订购系统中的事件传输效率和可靠性有着重要的意义.

参 考 文 献

[1] Eugster P T, Felber P A, Guerraoui R, et al. The many faces of publish/subscribe [J]. ACM Computing Surveys, 2003, 35(2): 114-131

[2] Ma Jiangang, Huang Tao, Wang Jinling. Underlying techniques for large-scale distributed computing oriented publish/subscribe system [J]. Journal of Software, 2006, 17(1): 134-147 (in Chinese)
(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核心技术[J]. 软件学报，2006，17(1)：134-147)

[3] Carzaniga A, Rosenblum D S, Wolf A L. Design and evaluation of a wide-area event notification service [J]. ACM Trans on Computer Systems, 2001, 19(3): 332-383

[4] Gough K J, Smith G. Efficient recognition of events in distributed systems [C] //Proc of the 18th Australasian Computer Science Conf. Los Alamitos, CA: IEEE Computer Society, 1995

[5] Aguilera M K, Strom R E, Sturman D C, et al. Matching events in a content-based subscription system [C] //Proc of the 18th ACM Symp on Principles of Distributed Computing. New York: ACM, 1999: 53-61

[6] Campailla A, Chaki S, Clarke E, et al. Efficient filtering in publish-subscribe systems using binary decision diagrams [C] //Proc of the ICSE 2001. Los Alamitos, CA: IEEE Computer Society, 2001: 443-452

[7] Yan T, Garcia-Molina H. Hector Garcia-Molina. Index structures for selective dissemination of information under the Boolean model [J]. ACM Trans on Database Systems, 1994, 19(2): 332-364

[8] Carzaniga A, Wolf A L. Forwarding in a content-based network [C] //Proc of ACM SIGCOMM 2003. New York: ACM, 2003: 163-174

[9] Xue Tao, Feng Boqin. Efficient matching for content-based publish-subscribe systems [J]. Mini-Micro Systems, 2006, 27(3): 529-533 (in Chinese)
(薛涛，冯博琴. 基于内容的发布订购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)：529-533)

[10] Altinel M, Franklin M J. Efficient filtering of XML documents for selective dissemination of information [C] // Proc of the 26th Int Conf on Very Large Data Bases. San Francisco: Morgan Kaufmann, 2000: 53-64

[11] Diao Y, Altinel M, Franklin M J, et al. Path sharing and predicate evaluation for high-performance XML filtering [J]. ACM Trans on Database Systems, 2003, 28(4): 467-516

[12] Chan C Y, Felber P, Garofalakis M, et al. Efficient filtering of XML documents with XPath expressions [J]. The VLDB Journal, 2002, 11(4): 354-379

[13] Peng F, Chawathe S S. XPath queries on streaming data [C] //Proc of the ACM SIGMOD Int Conf on Management of Data. New York: ACM, 2003: 431-442

Liu Guo, born in 1983. Received his master degree in computer science from Beihang University. His main research interests include distributed virtual reality.
刘 国，1983 年生，硕士，主要研究方向为分布式虚拟现实.

Zhou Zhong, born in 1978. PhD, associate professor, and member of China Computer Federation. His main research interests include virtual reality and distributed simulation.

周　忠,1978 年生,博士,副教授,中国计算机学会会员,主要研究方向为虚拟现实与分布式仿真等(zz@vrlab. buaa. edu. cn).

Wu Wei, born in 1961. Professor with doctor degree and PhD supervisor. Senior member of China Computer Federation. His main research interests include virtual reality, distributed system and network technologies.

吴　威,1961 年生,博士,教授,博士生导师,中国计算机学会高级会员,主要研究方向为虚拟现实、分布式系统、网络技术等.

Research Background

The event matching approach plays an important role in Map based publish/subscribe systems, the efficiency of the matching algorithm is a key point to evaluate the matching approach. In this paper, concentrating on how to reduce the matching times between attributes and constraints, we introduce a new matching approach based on the judgment of redundant attributes. By eliminating the redundancy of the attributes and maintaining the multilevel index structure of the constraints, it enhances the event matching efficiency and maintainability.

Our work is supported by the National 973 Program (2009CB320805) and the China Next Generation Internet Program (CNGI2008-123) and the Fundamental Research Funds for the Central Universities of China.

发布/订购系统中基于重复属性判定的事件匹配算法研究

作者： 刘国， 周忠， 吴威， Liu Guo， Zhou Zhong， Wu Wei
作者单位： 刘国,Liu Guo(虚拟现实技术与系统国家重点实验室(北京航空航天大学),北京,100191)，
周忠,Zhou Zhong(北京航空航天大学计算机学院,北京,100191)， 吴威,Wu Wei(中国计算机学会)

刊名： 计算机研究与发展 ISTIC EI PKU
英文刊名： JOURNAL OF COMPUTER RESEARCH AND DEVELOPMENT
年，卷(期)： 2010,47(10)
被引用次数： 1次

参考文献(13条)

1. Eugster P T;Felber P A;Guerraoui R The many faces of publish/subscribe[外文期刊] 2003(02)

2. 马建刚;黄涛;汪锦岭 面向大规模分布式计算发布订阅系统核心技术[期刊论文]-软件学报 2006(01)

3. Carzaniga A;Rosenblum D S;Wolf A L Design and evaluation of a wide-area event notification service[外文期刊] 2001(03)

4. Gough K J;Smith G Efficient recognition of events in distributed systems 1995

5. Aguilera M K;Strom R E;Sturman D C Matching events in a content-based subscription system[外文会议] 1999

6. Campailla A;Chaki S;Clarke E Efficient filtering in publish-subscribe systems using binary decision diagrams[外文会议] 2001

7. Yan T;Garcia-Molina H Hector Garcia-Molina.Index structures for selective dissemination of information under the Boolean model 1994(02)

8. Carzaniga A;Wolf A L Forwarding in a content-based network 2003

9. 薛涛;冯博琴 基于内容的发布订购系统中快速匹配算法的研究[期刊论文]-小型微型计算机系统 2006(03)

10. Altinel M;Franklin M J Efficient filtering of XML documents for selective dissemination of information[外文会议] 2000

11. Diao Y;Altinel M;Franklin M J Path sharing and predicate evaluation for high-performance XML filtering[外文期刊] 2003(04)

12. Chan C Y;Felber P;Garofalakis M Efficient filtering of XML documents with XPath expressions[外文期刊] 2002(04)

13. Peng F;Chawathe S S XPath queries on streaming data 2003

本文读者也读过(9条)

1. 张翔.邓赵红.王士同.ZHANG Xiang.DENG Zhao-hong.WANG Shi-tong 具有更好适应性的间距最大化特征加权[期刊论文]-计算机应用2010,30(9)

2. 张利.仲崇权.郑德宜.杨素英.黄陵碧.Zhang Li.Zhong Chongquan.Zheng Deguan.Yang Suying.Huang Lingbi 基于MVCD模型的网上订购系统的开发[期刊论文]-计算机工程与应用2005,41(7)

3. 陈娟 基于混合模式的网上订购系统的探讨和研究[期刊论文]-商场现代化2009(11)

4. 王秀琳.曹云峰.WANG Xiu-lin.CAO Yun-feng 基于单片机的微型飞行器高度计[期刊论文]-传感器与微系统2006,25(5)

5. 赵家伟.沈建新.廖文和 基于.NET平台的B2B网上订购系统的分析和实现[期刊论文]-计算机应用研究2004,21(1)

6. 陈治平.李小龙.王雷.林亚平.蔡立军.Chen Zhiping.Li Xiaolong.Wang Lei.Lin Yaping.Cai Lijun 最佳匹配

问题的DNA表面计算模型[期刊论文]-计算机研究与发展2005,42(7)

7. 李坡.赵旭.付佳晖.付云鹏.LI Po.ZHAO Xu.FU Jia-hiu.FU YUN-PENG 力敏Z-元件与新型力数字传感器[期刊论文]-传感技术学报2005,18(2)

8. 苑洪亮.张捷.郭长国.史殿习.YUAN Hong-liang.ZHANG Jie.GUO Chang-guo.SHI Dian-xi 内容发布订阅系统中事件可靠传递的研究[期刊论文]-计算机工程与科学2007,29(9)

9. 林英彬 基于SJM的订购系统的设计与实现[学位论文]2006

引证文献(1条)

1.张冬梅.王磊 面向分布式网络的信息按需分层分发系统框架[期刊论文]-自动化仪表 2011(8)