ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.19, No.9, September 2008, pp.2191−2202
DOI: 10.3724/SP.J.1001.2008.02191

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

一种基于速率的发布/订阅系统的准入控制机制*

郭祥丰[1,2+], 钟 华[1], 张文博[1,2], 李 京[1]

[1](中国科学院 软件研究所 软件工程技术中心,北京 100190)

[2](中国科学院 研究生院,北京 100049)

A Rate-Based Admission Control Scheme for Content-Based Publish/Subscribe Systems

GUO Xiang-Feng[1,2+], ZHONG Hua[1], ZHANG Wen-Bo[1,2], LI Jing[1]

[1](Technology Center of Software Engineering, Institute of Software, The Chinese Academy of Sciences, Beijing 100190, China)

[2](Graduate University, The Chinese Academy of Sciences, Beijing 100049, China)

+ Corresponding author: E-mail: wylde@otcaix.iscas.ac.cn

Guo XF, Zhong H, Zhang WB, Li J. A rate-based admission control scheme for content-based publish/subscribe systems. *Journal of Software*, 2008,19(9):2191−2202. http://www.jos.org.cn/1000-9825/19/2191.htm

Abstract: This paper presents RacsCBPS, an admission control scheme for large-scale and scalable content-based publish/subscribe systems. First the key requirements to implement admission control in content-based publish/subscribe systems are identified and how it differs from admission control schemes in the Internet and other research areas is analyzed. A cover-relation based algorithm to compute subscription resource requirements and an admission control algorithm based on subscription routing are presented. The scheme ensures time, space and control decoupling without sacrificing scalability of publish/subscribe systems. Publish/Subscribe systems can seek different balances between system resource utilization and QoS guarantee by choosing different admission control criteria. Experimental results show the effectiveness of the method.

Key words: publish/subscribe; quality of service guarantee; admission control; overload management

摘 要: 提出了基于内容发布/订阅系统的一种准入控制机制 RacsCBPS 来保障客户端的服务质量.首先分析了基于内容发布/订阅系统中实现准入控制机制的难点,给出了 RacsCBPS 的基本模型,在此基础上,提出了基于覆盖关系的资源需求计算方法和以订阅分发路径为基础的准入控制算法.RacsCBPS 在不影响发布/订阅系统松散耦合性的情况下,能够有效地解决因系统资源不足而导致的事件过载问题,同时为系统提供了不同的准入准则,以便在系统资源利用率和服务质量保证程度之间作出不同的权衡.最后通过实验验证了准入控制机制的有效性和相应的系统资源利用率.

* Supported by the National Natural Science Foundation of China under Grant Nos.60573126, 90718033 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant Nos.2003AA413010, 2003AA115440, 2006AA01Z180, 2007AA01Z134 (国家高技术研究发展计划(863)); the National Basic Research Program of China under Grant No.2002CB312005 (国家重点基础研究发展计划(973)); the National Key Technology Research and Development Program of China under Grant No. 2006BAH02A01 (国家科技支撑研究发展项目)

Received 2006-12-18; Accepted 2007-04-26

关键词: 发布/订阅;服务质量保证;准入控制;过载管理
中图法分类号: TP311 文献标识码: A

发布/订阅(publish/subscribe)是分布式系统中的一种通信范例,它为分布式应用提供了时间、空间、控制流3个方面的松散耦合性[1],这使得应用程序开发者更加有效地构建异构、高度动态、松散结合的应用.基于内容的发布/订阅系统(content-based publish/subscribe system)以其灵活的订阅机制成为发布/订阅系统的主流.目前,基于内容发布/订阅系统上的分布式应用越来越多,其中包括在线游戏[2]、在线股票报价[3]、RSS Feed[4]、安全警报服务[5]、位置服务[6]等.

在基于内容发布/订阅系统上的应用中,存在着大量的订阅者和发布者,典型应用的发布者数量在 10^2~10^5之间,订阅者数量在10^3~10^6之间[2−6].不同的客户端对于系统有着不同的服务质量需求,因此,如何保障应用的服务质量成为影响系统应用的一个关键性因素[7−9].然而,当过多的订阅者和发布者加入到系统中时,由于系统资源有限,可能会出现事件过载问题[10],造成事件的丢失,导致系统无法正常满足应用的需求.目前,对于发布/订阅系统中事件过载的解决方案大都采取尽力而为的方式[10,11],这使得系统无法为应用提供有保障的服务.对于由过多的请求导致系统无法保障应用的服务质量问题,一般引入准入控制机制进行解决.

准入控制是网络协议和 Web 服务器等研究领域的一个研究热点,很多准入控制策略和算法都被提了出来,然而,这些策略和算法大都是基于端到端的显式连接,且以集中的方式实现准入控制机制.相比之下,在基于内容发布/订阅系统中实现准入控制机制有以下难点:(1) 在系统中,发布者发布的事件被分发到多个订阅者,发布者和订阅者以一对多的方式进行通信;(2) 系统分发事件时,没有明确的分发路径,根据事件的内容来决定实际分发的路径;(3) 由于系统构建于一个分布式代理节点网络之上,集中的准入控制机制容易在系统中的某一节点上形成瓶颈,影响系统的伸缩性.

针对以上分析,要在基于内容发布/订阅系统中实现准入控制,需要解决以下几个方面的关键需求:(1) 系统需要能在事件无明确分发路径的条件下根据订阅请求计算该订阅对系统资源的需求;(2) 系统应保证应用的松散耦合性,保证订阅者和发布者在空间、时间、控制流方面的松散耦合;(3) 系统需要以一种非集中的方式实现准入控制机制.

在本文中,我们提出了一种基于速率的准入控制机制 RacsCBPS(rate-based admission control scheme for content-based publish/subscribe).RacsCBPS 是一种分布式准入控制机制,它基于过滤条件之间的覆盖关系,在订阅和公告分发的同时进行资源需求计算,同时采用基于节点的准入控制策略,订阅的准入控制通过一种基于订阅分发反向路径的算法来实现.

本文第 1 节比较相关研究工作.第 2 节给出系统模型.第 3 节给出基于覆盖关系的订阅资源需求计算方法.第 4 节给出以订阅分发路径为基础的准入控制算法.第 5 节给出验证 RacsCBPS 有效性的实验和相关分析评价.第 6 节对全文进行总结.

1 相关工作

按照事件的组织和描述形式的不同,发布/订阅系统分为基于主题、基于类型和基于内容 3 类[1],目前的主流研究集中于基于内容的发布/订阅系统.有关基于内容的发布/订阅的研究又分为功能性方面和非功能性方面的研究.非功能性方面的研究成为近年来发布/订阅系统的研究热点.非功能性方面的研究主要是针对发布/订阅系统的服务质量方面[7−9],其中包括网络延迟[12,13]、可靠性保证[14]等方面.

文献[8]提出了发布/订阅系统准入控制的概念,但是没有给出相关的解决方案.针对由于事件过载而给出的解决方案大都是基于尽力而为的方式,不能为应用提供有保障的服务.它们在事件过载的情况下,选择将事件丢弃或者长时间放置于等待队列中,这样会最终导致事件丢失或无效.文献[10]给出了类似网络协议拥塞处理的控制机制来解决发布/订阅系统中的事件过载,它根据系统的资源来降低系统中订阅者的接收速率和发布者的发布速率,并不限制过多的发布者或订阅者参与到系统中来.当系统发生事件过载时,发布者和订阅者无法按照

正常的速率和合理的延迟接收和发布事件.文献[11]针对外部环境或系统内部资源不足而引起的事件过载,给出了事件在分发过程中的基于市场理论的选择策略,但是当事件过载时,采取的是丢弃事件的做法,同样无法保证应用的服务质量.文献[7]给出了一种具有 QoS 支持的基于 DHT 的发布/订阅系统 IndiQos,但是没有给出如何对系统中提交的订阅进行资源需求计算.它要求应用给出订阅所需的资源需求放置于订阅请求中,但是,仅仅根据订阅请求无法计算出订阅所需的网络和计算能力等系统资源.另外,IndiQos 也没有给出相关的准入控制算法.

在针对由于系统资源不足而导致无法为用户提供有保障服务的问题上,网络协议和中间件领域提出了很多解决方案,这些解决方案主要分为两类:一类是提供“硬”的保障,提出的相应协议和解决机制能够完全保证用户的服务,这类解决方案要求服务网络或节点能够预留资源,同时要维护请求或连接的相关状态.这类解决方案能够严格地保证服务质量,但是资源利用率不能很好地保证;另外一类解决方案在一定程度上保证了服务质量,提供“软”的保证.Differentiated Services (DiffServ)就是其中的一类解决方法.DiffServ 不要求对每个请求进行控制,也不要求相应的信号机制,系统不需要维护每个请求的状态,而仅仅实现优先级调度和缓存机制.在这种方式下,当系统发生过载时,所有的请求都将会受到影响,导致服务降级.DiffServ 的目的并不是针对单个请求提供有保证的服务,而是从系统的全局进行考虑.本文提出的方法使得用户既能选择“硬”的保证,又可以选择“软”的保证,以提高系统的资源利用率.

在网络协议等其他研究领域,有关准入控制方面的研究很多[15−20],在准入准则方式上主要分为基于速率和基于延迟两种方式,在控制方式上主要分为基于参数和基于度量两种.但是,它们大都是基于明确路由的一对一通信的解决方案.相比之下,基于内容发布/订阅系统采用一对多的通信方式,根据事件内容确定路由,这些解决方案并不适用于基于内容的发布/订阅系统.而本文针对这些不同之处采取了基于速率的准入准则,在控制方式上采用基于度量的方式,在不影响系统的松散耦合性和伸缩性的情况下,给出了相应的解决方案.

2 系统模型

为了更好地描述基于内容发布/订阅系统的准入控制机制,我们首先给出基于内容发布/订阅系统的模型,然后给出 RacsCBPS 的基本组成部分.

2.1 基于内容发布/订阅系统模型

一个发布/订阅系统由代理节点组成的网络、一组发布者和一组订阅者构成,其中,每个发布者或订阅者都连接到代理网络的某一节点上(如图 1 所示).发布者以事件的形式将信息发布到代理网络中;订阅者通过发出订阅请求表示对特定的事件感兴趣;代理网络则负责将事件及时、可靠地分发给所有感兴趣的订阅者.

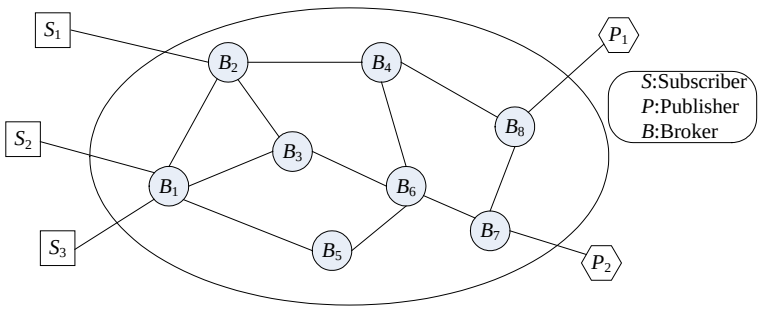


Fig.1 Model of publish/subscribe systems

图 1 发布/订阅系统模型

为了后文的描述与分析,我们给出如下定义,用于系统模型的刻画.

定义 1. 一个事件 e 是一个属性的集合$\{a_1,a_2,\ldots,a_n\}$,其中,a_i 是一个名字值对(n_i,v_i),$i\neq j\Rightarrow n_i\neq n_j$.

定义 2. 一个属性过滤条件 c 是一个三元组(n,op,V),n 是属性名字,op 是一个操作符,V 是一个属性值的集合.

定义 3. 一个属性过滤函数 $f(a)$是一个布尔函数,$f(a)$=true 当且仅当属性 a 满足 f 指定的属性过滤条件 c.

定义 4. 一个过滤条件是一个由属性过滤函数集合$\{f_1,f_2,\ldots,f_m\}$组成的布尔函数 $F(e)=f_1\wedge f_2\wedge\ldots\wedge f_m$,$F(e)$=true 当且仅当对于事件 e 中的任一属性 a_i,存在一个 $f_j\in\{f_1,f_2,\ldots,f_m\}$使得 $f_j(a_i)$=true.所有满足过滤条件 F_i 的事件集合记做 E_i,$E_i=\{e|F_i(e)=\text{true}\}$.

定义 5. 过滤条件 F_i 覆盖 F_j(记做 $F_i\supset_c F_j$),当且仅当 $E_i\supset E_j$.

过滤条件之间的覆盖关系$\supset_c$为一种偏序关系(证明略).

定义 6. 如果 $F_i\supset_c F_j$,并且不存在 F_k 使得 $F_i\supset_c F_k$,$F_k\supset_c F_j$,则称 F_i 直接覆盖 F_j.所有直接覆盖 F_j 的过滤条件构成了 F_j 的前序覆盖集.所有被 F_j 直接覆盖的过滤条件构成了 F_j 的后序覆盖集.

代理网络中的每个节点维护着订阅(subscription)和公告(advertisement)信息,这些信息利用过滤条件之间的覆盖关系进行简化,组成树状结构,形成订阅关系树或公告关系树.覆盖关系是一种偏序关系,因此,订阅关系树和公告关系树都是有向无环图.

基于内容发布/订阅系统中的参与者通过以下接口(见表 1)进行交互,其中,发布者使用的接口包括 *publish*(*event*),*advertise*(*filter*)和 *unadvertise*(*filter*),代理网络节点使用的接口为 *notify*(*event*),订阅者使用的接口包括 *subscribe*(*filter*)和 *unsubscribe*(*filter*);接口中的 *filter* 参数表示过滤条件.

Table 1 Typical interface functions and their semantics in a publish/subscribe system

表 1 基于内容发布/订阅系统用户接口及其操作语义

User interface	Semantics
publish(*event*)	Used by a publisher to publish an event
subscribe(*filter*)	Used by a subscriber to express its interested events
unsubscribe(*filter*)	Used by a subscriber to cancel its interest on specific events
notify(*event*)	Used to notify the interested subscribers upon the arrival of an event
advertise(*filter*)	Used by a publisher to advertise the events it publishes
unadvertise(*filter*)	Used by a publisher to cancel an advertisement

2.2 准入控制机制

RacsCBPS 由系统资源管理、订阅资源请求、准入控制算法和准入控制准则等几个部分组成(如图 2 所示).

其中,系统资源管理提供了对系统可用资源的记录和管理;订阅资源请求主要用于计算新的订阅所需的系统资源;准入控制准则为准入控制算法提供了拒绝或者接纳一个请求的条件.准入控制算法根据系统可用资源、订阅所需系统资源和准入控制准则进行准入控制决策,决定系统是否接纳指定的订阅.

在以下几节里,我们将分别介绍 RacsCBPS 的各个部分.

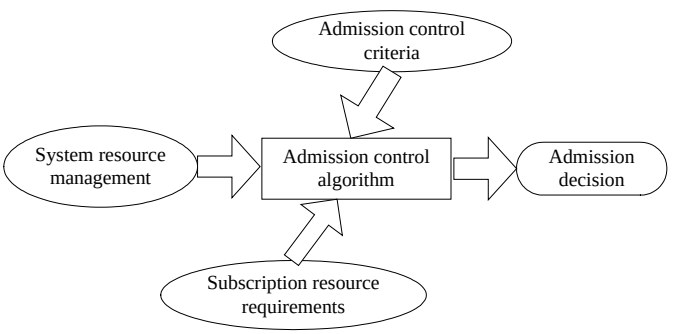


Fig.2 Relationship between basic components of RacsCBPS

图 2 RacsCBPS 相关组成部分及其关系图

3 订阅的资源需求计算

RacsCBPS 的数据模型和路由算法是建立在 SIENA[21]的基础上的.当订阅者发布一个订阅时,系统需要对其所需的资源进行计算,从而决定是否接纳这个订阅.我们对订阅所需资源计算的思路是,对当前数据模型和用户接口进行扩展,通过发布者发布的公告对发布事件的特征进行描述.这些信息随着公告扩散到整个网络中,当订阅者发布一个新的订阅时,使用该订阅与系统中的公告关系树进行运算,得出该订阅所需的资源.

3.1 扩展公告

为了实现准入控制机制,需要对现有的公告进行扩展.在大多数应用中,发布者数量一般比订阅者数量少很多,而且比起订阅者来说相对稳定,不会频繁地退出和加入,发布事件的规律性比较强[2−6].当一个发布者加入到系统中来时,需要提供该发布者发布事件的最大速率 v_{pmax},如果发布者没有指定最大发布速率,则系统根据已有资源和发布者发布事件的历史数据给出该发布者发布事件的默认最大值,系统会控制发布者发布到系统中的最大速率,使其不会大于最大值 v_{pmax},从而在发布者一方实现准入控制.

我们对公告关系树中的每个节点 N 扩展以下控制信息:

(1) *count*,匹配该节点所表示订阅的事件来源发布者个数;

(2) *rate*,匹配该节点所表示订阅的事件的最大发布速率.

由于 RacsCBPS 是一种基于速率的准入控制机制,*rate* 属性作为描述事件流的指标,*count* 属性则作为实现扩展接口的一个辅助属性,同时也为系统扩展其他功能和实验统计提供便利.公告在代理网络中分发时,在各个代理节点中将会构建公告关系树,公告关系树中也将包括上述两类信息(如图 3 所示).

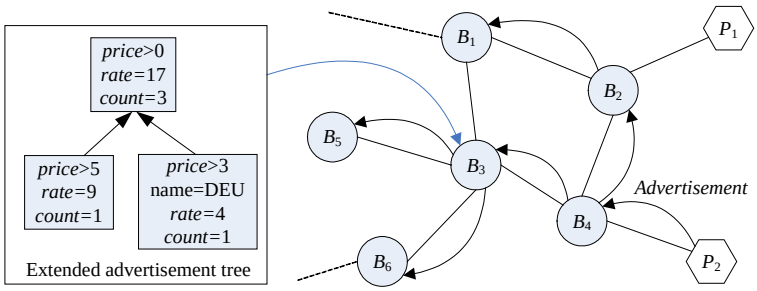


Fig.3 Extended advertisement in RacsCBPS

图 3 RacsCBPS 的扩展公告

3.2 *advertise*的实现

当发布者发布一个公告时,公告将会沿着以它所连接代理节点为根形成的一个最小生成树进行扩散,扩散的过程中更新各个节点的公告关系树.设公告所关联的过滤条件为 F,公告所声明的最大发布速率为 v_{pmax}.当该公告扩散到某一代理节点时,存在以下两种情况:

(a) 代理节点上的公告关系树中存在一个节点 $N=F$,这时更新 N 的控制信息 $N.count=N.count+1$,$N.rate=N.rate+v_{pmax}$.同时更新 N 的所有祖先节点,更新规则与更新 N 相同(如图 4(a)所示).由于所有祖先节点只累加 1 次,所以不会出现重复计算问题;

(b) 代理节点的公告关系树中不存在节点 N,使得 $N=F$,则在公告关系树中新加入一个节点 M,M 将插入到 F 的前序覆盖集 S 和后序覆盖集 T 之间,构成新的公告信息树.由于覆盖关系是一种偏序关系,因此,后序覆盖集 T 中的过滤条件所对应的事件集合之间可能存在相交的关系,即 T 中所有的过滤条件所对应的后序覆盖集设为 W,存在 $w_i,w_j\in W,w_i\cap w_j\neq\varnothing$,因此,如果直接进行后序覆盖关系集的控制信息相加,会出现重复计算问题,令 $Z=\bigcup_{w\in W} w$,重复计算的速率:

$$d_{rate} = \sum_{w \in W} \sum_{t \in w} t.rate - \sum_{s \in Z} s.rate .$$

重复计算的来源发布者个数:

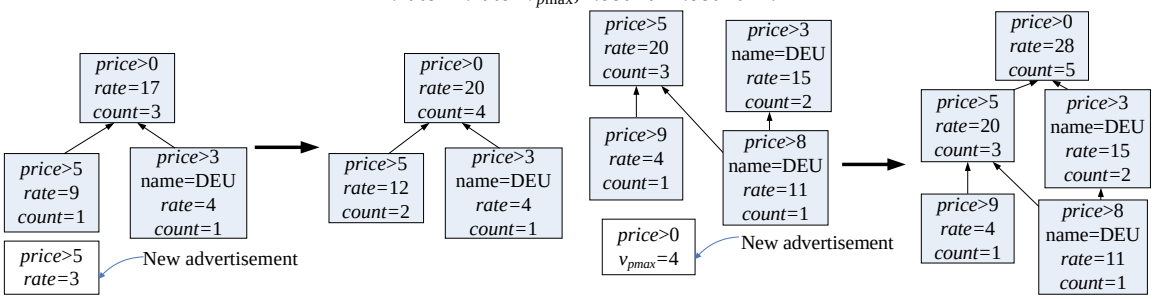
$$d_{count} = \sum_{w \in W} \sum_{t \in w} t.count - \sum_{s \in Z} s.count .$$

构建 M 的控制信息(如图 4(b)所示):

$$M.rate = v_{p\max} + \sum_{t \in T} t.rate - d_{rate}, \; M.count = 1 + \sum_{t \in T} t.count - d_{count} .$$

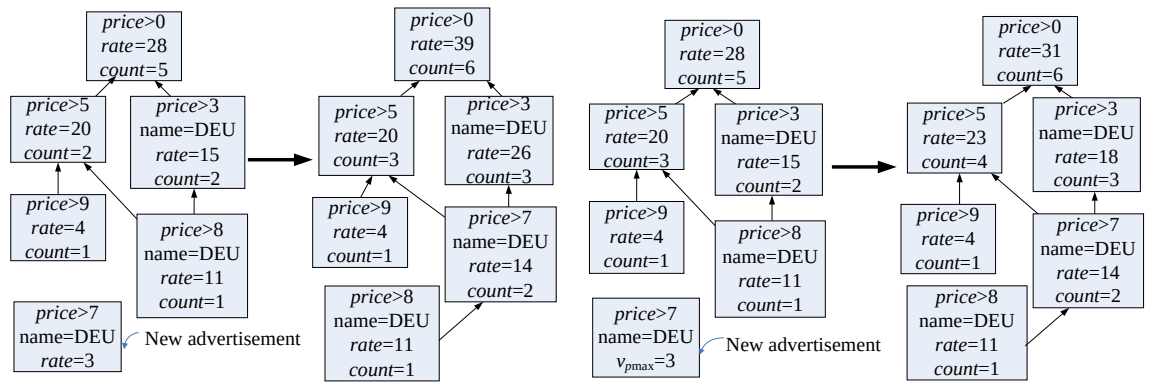
更新 M 的所有祖先节点(如图 4(c)所示),对于任一祖先节点 L,更新规则为

$$L.rate=L.rate+v_{p\max}, L.count=L.count+1.$$



(a) Update advertisement information in case same filter exists
(a) 存在相同过滤条件的公告更新

(b) Successor related calculating of control information
(b) 后序覆盖集的相关计算



(c) Predecessor related calculating of control information
(c) 前序覆盖集的相关计算

Fig.4 Examples of advertisement calculation for different cases
图 4 不同情况下的公告关系树更新示意图

3.3 *unadvertise*操作的实现

当一个发布者发布一个 *unadvertisement*(取消公告)时,该 *unadvertisement* 沿着以该发布者所连接代理节点为根形成的最小生成树进行公告关系树的更新.设 *unadvertisement* 中的过滤条件为 F,对应的 *advertisement* 的最大发布速率为 $v_{p\max}$,这时,需要对每个节点中的公告关系树进行更新.在公告关系树中,必定存在一个节点所表示的过滤条件 $N=F$,更新该公告关系树时,存在以下两种情况:

(a) 如果 $v_{p\max}<N.rate,N.count>1$,说明发布此类型事件的发布者有多个,则对它的控制信息进行更新,同时,更新它的所有祖先节点,计算过程类似 *advertise* 操作;

(b) 如果 $v_{p\max}=F.rate$,则首先更新它的所有祖先节点,然后删除这个节点.由于覆盖关系是偏序关系,它的前序覆盖集 S 中的任一过滤条件和后序覆盖集 T 中的任一过滤条件都存在偏序关系,建立它们之间的偏序关系.

3.4 订阅所需资源的计算

当一个订阅者发布一个订阅时,该订阅会在系统中进行扩散.当该订阅到达代理网络中的某一节点时,系统会根据该订阅的订阅条件 F 和此节点上的公告关系树对该订阅在此节点上所需资源进行计算,存在以下两种情况:

(a) 如果此节点的公告关系树中存在某一节点 N 与订阅条件 F 相同,则表示此订阅需要该节点能够以 $N.rate$ 的速率进行处理,它订阅的事件来自 $N.count$ 个发布者;

(b) 如果此节点的公告关系树中不存在与 F 相同的节点,则在公告关系树中存在 F 的前序覆盖集 S 和后序覆盖集 T,我们根据这两个集合计算此订阅在该节点所需资源的最大值和最小值(公式(1)).

$$\begin{cases} \min Rate = \max(t.rate), & t \in T \\ \max Rate = \min(s.rate), & s \in S \\ \min Count = \max(t.count), & t \in T \\ \max Count = \min(s.count), & s \in S \end{cases} \tag{1}$$

定理 1. 在 RacsCBPS 中,代理节点上满足过滤条件 F 的事件到达速率属于[min$Rate$,max$Rate$]区间.

证明:由于过滤条件之间存在的覆盖关系,对于前序覆盖关系集 S 中任一过滤条件 s 均覆盖 F,满足过滤条件 F 的事件集合 E_F 是满足过滤条件 s 的事件集合 E_s 的子集,因此,E_F 的事件生成速率小于 E_s 的生成速率,E_F 事件生成速率的上限为 min($s.rate$).同理,E_F 事件生成速率的下限为 max($t.rate$).证毕. □

如果订阅者发布的一个订阅 *subscribe*(“*price*>2,name=DEU”)所到达的代理节点上的公告关系树如图 4(c)所示,则可以计算出该订阅需要此节点能以最小不低于 18,最大不高于 31 的速率进行处理(如图 5 所示),该订阅所订阅事件至少来自 3 个发布者,但不会超过 6 个发布者.

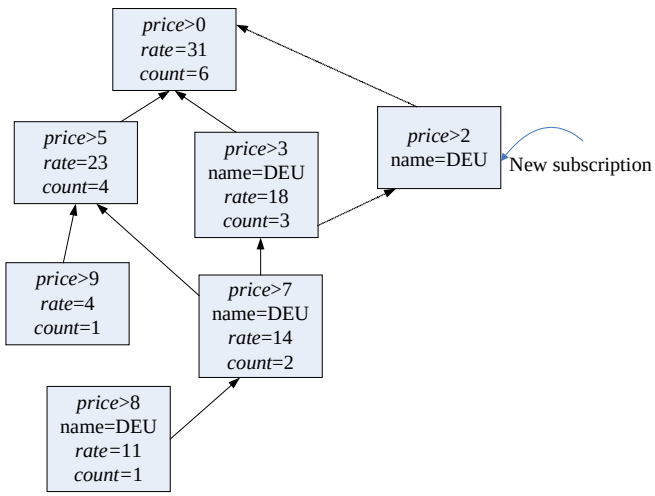


Fig.5 Subscription resource requirements computation

图 5 新的订阅所需资源的计算

3.5 资源需求计算复杂度分析

订阅所需资源需求的计算主要涉及计算过滤条件之间的覆盖关系,计算覆盖关系的时间复杂度为 $O(n^2)$,其中,n 为过滤条件中属性的个数.设 N 为公告关系树中节点的个数,在公告关系树进行更新时,需要将新的公告与公告关系树中的每个节点进行覆盖关系计算,公告关系树更新的时间复杂度为 $O(N)$.在 *advertise* 操作的两种情况下,对控制信息的更新的时间复杂度也为 $O(N)$.实际上,控制信息的计算与公告关系树的更新是同时进行的.同样,*unadvertise* 操作的时间复杂度也为 $O(N)$;*subscribe* 操作返回控制信息计算的时间复杂度为 $O(N)$.因此,与基本的订阅和公告分发算法[21]相比,本文中方法并没有增加计算的复杂度.

4 准入控制算法

准入控制机制根据系统的可用资源、准入控制准则以及准入控制算法对系统中新的订阅进行计算.在这一部分中,我们首先给出系统中的资源管理,然后讨论代理节点的准入控制准则,最后给出准入控制算法.

4.1 代理节点上的资源管理

每个代理节点管理着各自的可用资源,为准入控制机制提供计算依据.RacsCBPS 主要针对几种关键资源进行管理,其中包括网络带宽、CPU 计算能力和内存.由于我们使用覆盖关系进行计算,各节点对内存的需求与节点上的订阅关系和公告关系的复杂度成正比,计算比较简单,我们在此不作详细说明.各节点间的网络带宽一般是固定值,可以通过资源预留协议(RSVP)来保证.

每个节点记录着 CPU 达到极限值时的事件最大处理速率,初始值为 0.在系统运行过程中,通过每隔一个时间间隔 I 进行采样;当系统侦测到 CPU 利用率达到某一极限值时,记下当前的事件处理速率,对此速率 R_{new} 与已记录事件最大处理速率 $R_{\max}$ 进行比较,选取它们之中的最大值 $R'_{\max}=\max(R_{new},R_{\max})$ 作为新的事件最大处理速率.RacsCBPS 以最大事件处理速率来表示 CPU 资源的处理能力.

4.2 节点准入准则

与网络协议中的准入控制类似,描述一个节点的事件流可以采用两种方式:基于峰值和基于平均值的方式[19,20].为了更加严格地保证客户端的服务质量,RacsCBPS 采用基于峰值的方式[19].当一个订阅到达某一代理节点时,该代理节点根据第 3 节中给出的方法进行订阅的资源需求计算,得出该订阅的资源需求;系统根据资源需求信息进行准入控制.我们使用基于峰值的计算方法,如果该代理节点无法满足事件峰值时的处理速率,则拒绝该订阅.

如果该节点无法满足以下任一条件,则拒绝该订阅:

(a) 该订阅的事件处理速率加上节点所连接路径中的已有事件处理速率小于节点所连接路径的预留网络带宽;

(b) 各路径中需要的节点事件处理速率小于节点上 CPU 的最大事件处理速率;

(c) 订阅和公告所需的内存小于节点上能分配内存的最大值.

4.3 算法描述

当订阅者发布一个订阅请求时,RacsCBPS 通过执行准入控制算法决定是否接纳这个订阅,准入控制算法的流程如图 6 所示.

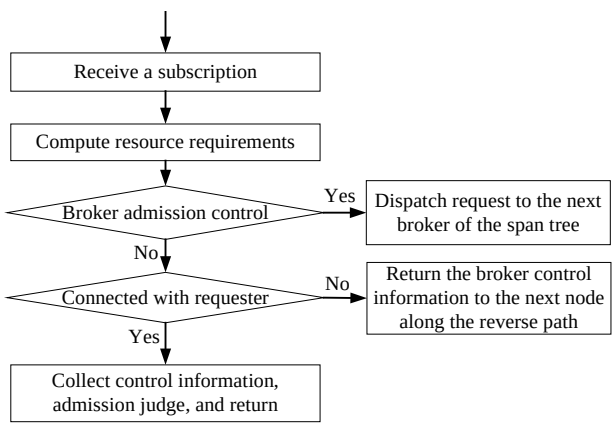


Fig.6 Control flow of admission control algorithm

图 6 准入控制算法流程图

准入控制算法(如图 7 所示)执行以下步骤:

1.当订阅请求扩散到系统中各代理节点时,在加入到各节点订阅关系树之前,根据节点准入准则决定是否接纳此订阅:如果节点接纳此订阅,则此订阅继续向前分发;否则,根据不同的准入准则决定是否继续分发.各节点将沿着订阅分发的反向路径,一直将拒绝信息传递到订阅者连接的代理节点(如图 8 所示).

2.订阅者所连接的节点根据沿着订阅分发逆向路径反馈回来的控制信息来决定是否接纳这个订阅.

```
//   Admission control-RacsCBPS
MessageConsumerImpl implements MessageConsumer
{   //subscriber implementation
    …
    //subscribe(filter) implementation
    public void onSubscribe(Subscritption sub) throws
            NotAccpetedException
    // input: Subscription including specific filter
    // output: if allow return void, otherwise throw Exception
    // use global variable PERCENTAGE_GUARANTEED
    {
        // compute subscription resource requirements
        requiredRes=calculateWithAdvertisementInfo(sub.filter);
        // according to local resource, make admission decision
If (!acceptWithMemRes(requiredRes)||!acceptWithCPURes
    (requiredRes)||!acceptWithBandWidthRes(requiredRes))
    {
        // if percentage-based
        if (PERCENTAGE_GUARANTEED)
            // collect reject control information
            nextBroker.forwardSubscription(sub,
                    new NotAcceptedException(this));
        else // fully guaranteed policy
             // need not to collect all control information
             throws new NotAcceptedException(this);
        ….
    }
}
```

Fig.7 Admission control algorithm

图 7 准入控制算法

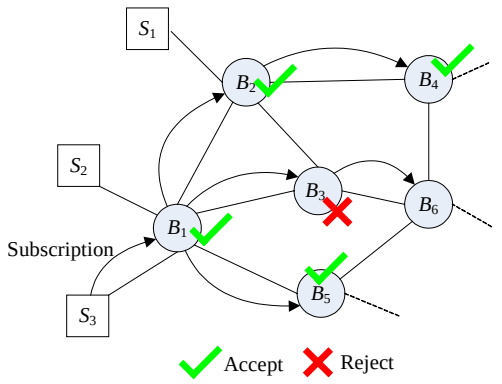


Fig.8 Illustration of admission control algorithm

图 8 准入控制算法示意图

基于收集到的来自各个节点的控制信息,RacsCBPS 针对不同的应用需求,给出以下两个准入控制准则:

(a) 完全保证:只要有一个节点拒绝了该订阅,则系统拒绝该订阅.如图 8 所示,订阅者 S_3 发出一个订阅请求,收到一个来自节点 B_3 的拒绝消息,则系统拒绝这个订阅.这种准入控制准则提供了一种“硬”的保证.在这种情况下,能够完全保证订阅的事件处理速率,但是由于系统单个节点形成瓶颈的可能性比较高,导致订阅被拒绝的可能性也比较高,因此,系统资源利用率相对较低.这种准入控制准则适用于对系统要求比较严格的应用.

(b) 基于百分比:当收到的拒绝信息在所有收到的控制信息中,超过一定比例时,系统才拒绝该订阅.准入控制算法采取这种准则时,会收集所有节点上的控制信息,在算法第 1 步中,即使某一节点拒绝该订阅,订阅仍然沿着最小生成树继续分发.该准则不能完全保证订阅的事件处理速率,它在一定程度上保证了事件的处理速率,提高了系统资源的利用率.

4.4 算法复杂度分析

由于订阅的过滤条件之间也是覆盖关系,更新订阅关系树的规则与公告关系树的更新相同,因此,订阅关系

树更新的时间复杂度也为 $O(N)$,其中,N 为公告关系树中节点的个数.公告和订阅在代理网络中分发计算的时间复杂度为 $O(m)$,其中,m 为代理网络中节点的个数.准入控制算法需要在整个代理网络中更新各节点的控制信息,算法的时间复杂度为 $O(m)$.

5 实验及分析评价

5.1 实验设计

RacsCBPS 已经在中国科学院软件研究所的 Web 应用服务器产品 OnceAS[22]上进行了原型实现.本节通过比较无准入控制和应用准入控制机制的条件下,系统对应用的服务质量保障情况来验证 RacsCBPS 的有效性.

我们设置代理节点 50 个,发布者数量 20 个,订阅者数量 1~100 个,发布者和订阅者随机地分布在各个代理节点上.为了模拟节点的计算能力,我们限制不同的节点的事件处理速率以及节点之间的带宽,随机地限制每个节点处理事件的个数为 100 个/s~200 个/s 来模拟每个节点的 CPU 处理能力,节点之间的网络带宽随机地限制为 200 个/s~300 个/s 来模拟节点之间的带宽.发布者以 20 个事件/s~30 个事件/s 的速率发布随机生成的事件.为了保证实验的可对比性,随机生成一个固定的事件集合,每次发送者发送事件时,将从这个集合中选取一个事件,先执行一个 *advertise* 操作,然后发送事件 100 次,最后调用 *unadvertise*,不断地重复上述行为.同样,订阅者将不断地重复以下序列的动作:*subscribe,receive,receive,…,receive,unsubscribe*.当一个事件在一个节点超过 5s 没有被处理时,系统丢弃这个事件.

对于应用的服务质量保障,本文主要从系统能否及时地处理事件的角度来进行考察.当系统不能及时地处理系统中的事件时,会因超时将事件丢弃.我们以不同情况下各个节点中事件的丢弃率和各节点的处理能力和网络带宽的利用率,考察准入控制算法的有效性.由于在准入控制算法中,当节点拒绝信息超过 50%时,系统中多数路径将会难以保证系统的服务质量,因此,我们选取小于 50%的拒绝准则来验证非完全保证准入准则下准入控制算法的有效性,我们分别选取 20%拒绝准则和 40%拒绝准则.

5.2 结果分析

通过图 9 我们可以看出,当无准入控制机制时,随着订阅者的增多,平均每个节点上丢弃的事件会越来越多;而引入控制机制后,节点上的事件丢弃率会有明显的降低;当采取完全保证的准入控制准则时,系统中节点上的事件丢弃速率接近 0.当允许出现部分节点拒绝的情况下,20%拒绝准入准则的情况下,系统中的丢弃速率约为 50 个/秒;在 40%拒绝准入准则的情况下,当订阅者增加到 80 个以上时,系统中每个节点的丢弃率上升较快.完全保证的准入准则在系统达到一定负荷时,订阅的拒绝率很快就上升到 40%以上.相比之下,当采取 20%拒绝准入准则的情况下,订阅的拒绝率仍然比较低,但是采取 40%拒绝准入准则时,系统的拒绝率相对较低.

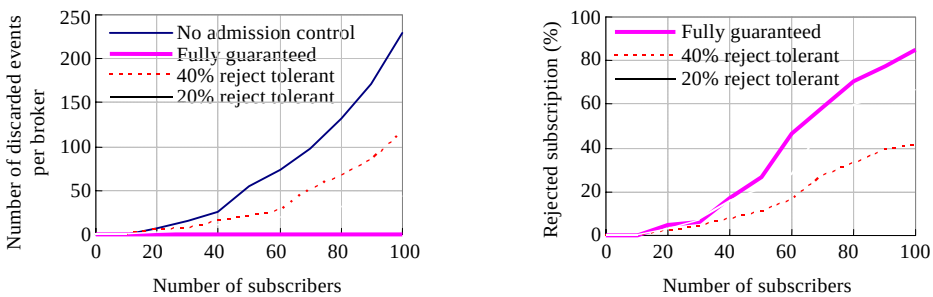


Fig.9 Events discarding rates and subscription reject rates of different criteria

图 9 不同准入准则下的节点平均事件丢弃率和订阅拒绝率

通过图 10 可以看出,系统在不同准入准则下的 CPU 利用率和内存利用率的情况.CPU 的利用率较内存的利用率高,当采取完全保证的准入控制准则时,系统中的 CPU 和内存利用率均较低.由此可以看出,如果采取较

为严格的准入控制准则,则整个系统的利用率比较低;而采取相对宽松的准入控制准则,则可以提高系统资源的利用率,但是在保证系统客户端的服务质量方面较弱.

因此,在应用对服务质量要求严格的情况下,可以采取相对严格的完全保证的准入控制准则;而对服务质量要求相对宽松的条件下,我们可以采取基于百分比的准入控制准则,在一定程度上保证了服务质量,还同时提高了系统资源的利用率.

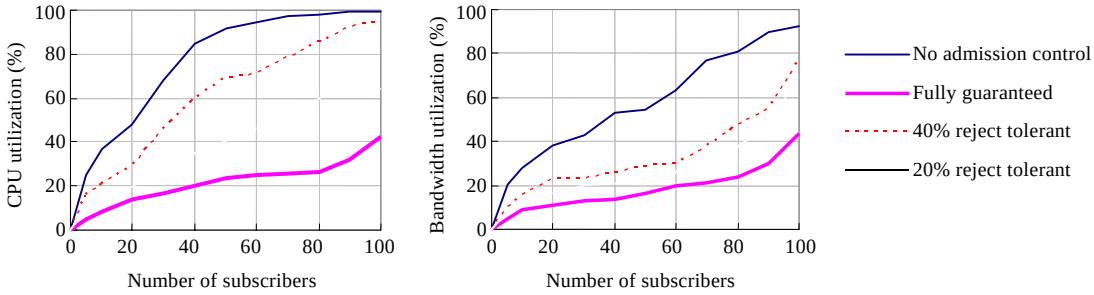


Fig.10 CPU and bandwidth utilization of different strategies

图 10 不同准入准则下的节点 CPU 利用率和网络带宽利用率

6 结束语

随着基于内容发布/订阅系统应用的不断增加,为这些应用提供可靠的服务质量保障成为衡量系统的一个重要因素.本文针对由于系统资源不足而引起的事件过载问题,提出了一种准入控制机制 RacsCBPS,RacsCBPS 基于过滤条件之间的覆盖关系进行订阅资源需求的计算,同时还给出了基于订阅分发的准入控制算法,算法提供两类不同的准入控制准则,使得不同的应用可以根据不同的服务质量需求选择不同的准入准则.最后,通过实验验证了准入控制算法的有效性.

本文的主要贡献在于:(1) 针对目前发布/订阅系统解决事件过载问题而采取的尽力而为方式的不足,提出了一种准入控制机制 RacsCBPS;(2) 解决了基于内容发布/订阅系统中无明确路由一对多通信方式下订阅请求的资源需求计算问题;(3) 提出了基于订阅分发路径的准入控制算法,为系统的应用提供了有保障的服务.

RacsCBPS 是一种基于速率的准入控制机制,我们的进一步工作将考虑如何在系统中实现基于延迟的准入控制机制,还将考虑如何在系统中实现服务差分,对不同的订阅者采取不同的服务质量保证.

References:

[1] Eugster PT, Felber PA, Guerraoui R, Kermarrec AM. The many faces of publish/subscribe. ACM Computing Surveys, 2003,35(2): 114−131.

[2] Farber J. Network game traffic modelling. In: Proc. of the 1st Workshop on Network and System Support for Games. Bruanschweig: ACM Press, 2002. 53−57.

[3] Wang YM, Qiu L, Achlioptas D, Das G, Larson P, Wang HJ. Subscription partitioning and routing in content-based publish/subscribe systems. In: Proc. of the 16th Int'l Symp. on Distributed Computing. Berlin: Springer-Verlag, 2002. 28−30.

[4] Liu H, Ramasubramanian V, Sirer EG. Client behavior and feed characteristics of rss, a publish-subscribe system for Web micronews. In: Proc. of the Internet Measurement Conf. Berkeley: USENIX, 2005. 29−34.

[5] Costa M, Crowcroft J, Castro M, Rowstron A, Zhou L, Zhang L, Barham P. Vigilante: End-to-End containment of Internet worms. ACM SIGOPS Operating Systems Review, 2005,39(5):133−147.

[6] Chen X, Chen Y, Rao F. An efficient spatial publish/subscribe system for intelligent location-based services. In: Proc. of the 2nd Int'l Workshop on Distributed Event-Based Systems. San Diego: ACM Press, 2003. 1−6.

[7] Carvalho N, Araujo F, Rodrigues L. Scalable Qos-based event routing in publish-subscribe systems. In: Proc. of the 4th IEEE Int'l Symp. on Network Computing and Applications. Washington: IEEE, 2005. 101−108.

[8] Araujo F, Rodrigues L. On QoS-aware publish-subscribe. In: Proc. of the 22nd IEEE Int'l Conf. on Distributed Computing Systems Workshops. Vienna: IEEE, 2002. 511−515.

[9] Behnel S, Fiege L, Muhl G. On quality-of-service and publish-subscribe. In: Proc. of the 26th IEEE Int'l Conf. on Distributed Computing Systems Workshops. Lisboa: IEEE, 2006. 20−25.

[10] Pietzuch PR, Bhola S. Congestion control in a reliable scalable message-oriented middleware. In: Proc. of the 4th Int'l Conf. on Middleware. Rio de Janeiro: Springer-Verlag, 2003. 202−221.

[11] Jerzak Z, Fetzer C. Handling overload in publish/subscribe systems. In: Proc. of the 26th IEEE Int'l Conf. on Distributed Computing Systems Workshops. Lisboa: IEEE, 2006. 32−37.

[12] McCaffery D, Finney J. Low latency optimisation of content based publish subscribe for real-time mobile gaming applications. In: Proc. of the 25th IEEE Int'l Conf. on Distributed Computing Systems Workshops. Columbus: IEEE, 2005. 438−443.

[13] Carvalho N, Araujo F, Rodrigues L. Reducing latency in rendezvous-based publish-subscribe systems for wireless ad hoc networks. In: Proc. of the 26th IEEE Int'l Conf. on Distributed Computing Systems Workshops. Lisboa: IEEE, 2006. 28−31.

[14] Bhola S, Strom R, Bagchi S, Zhao Y, Auerbach J. Exactly-Once delivery in a content-based publish-subscribe system. In: Proc. of the 32nd Annual IEEE/IFIP Int'l Conf. on Dependable Systems and Networks. Bethsda: IEEE, 2002. 7−16.

[15] Jain M, Dovrolis C. End to end available bandwidth: Measurement methodology, dynamics and relation with TCP throughput. IEEE/ACM Trans. on Networking, 2003,11(4):537−549.

[16] Breslau L, Knightly EW, Shenker S, Stoica I, Zhang H. Endpoint admission control: Architectural issues and performance. In: Proc. of the Sigcomm 2000. Stockholm: IEEE, 2000. 57−69.

[17] Perros HG, Elsayyed KM. Call admission control schemes: A review. IEEE Communication Magazine, 1996,34(11):82−91.

[18] Elek V, Karlsson G, Ronngren R. Admission control based on end-to-end measurements. In: Proc. of the IEEE INFOCOM 2000. Tel Aviv: IEEE, 2000. 623−630.

[19] Ferrari D, Verma D, A scheme for real-time channel establishment in wide-area networks. IEEE Journal on Selected Area in Communications, 1990,8(3):368−379.

[20] Lee T, Lai K, Duann S. Design of a real-time call admission controller for ATM networks. IEEE/ACM Trans. on Networking, 1996, 4(5):758−765.

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

[22] Huang T, Chen NJ, Wei J, Zhang WB, Zhang Y. OnceAS/Q: A QoS-enabled Web application server. Journal of Software, 2004, 15(12):1787−1799 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/15/1787.htm

附中文参考文献:

[22] 黄涛,陈宁江,魏峻,张文博,张勇.OnceAS/Q:一个面向 QoS 的 Web 应用服务器.软件学报,2004,15(12):1787−1799. http://www.jos.org.cn/1000-9825/15/1787.htm

郭祥丰(1980－)男,黑龙江北安人,博士生,主要研究领域为网络分布计算,软件工程.

张文博(1976－),男,博士生,主要研究领域为网络分布计算.

钟华(1971－),男,博士,研究员,CCF 高级会员,主要研究领域为软件工程,分布计算,对象技术,形式化方法.

李京(1966－),男,博士,研究员,博士生导师,主要研究领域为网络分布计算,软件体系结构,组合软件技术.