

大规模发布/订阅系统中的服务质量^{*})

董 彪¹ 陈金辉² 孙亚民¹

(南京理工大学计算机科学与技术学院 南京 210094)¹ (南京信息工程大学信息与控制学院 南京 210044)²

摘 要 发布/订阅系统已经获得广泛的关注,但在该领域仍然缺少意义明确的度量准则。全面地提出了相关的服务质量度量及其在发布/订阅系统中的语义,通过扩展系统操作,给出了带 QoS 约束条件的路由算法的一般准则,为进一步研究发布/订阅系统提供了坚实的基础。

关键词 服务质量,度量,质量受限的路由

Quality of Service in Large Scale Publish/Subscribe Systems

DONG Biao¹ CHEN Jin-hui² SUN Ya-min¹

(School of Computer Science & Technology, Nanjing University of Science & Technology, Nanjing 210094, China)¹

(School of Information & Control, Nanjing University of Information Science & Technology, Nanjing 210044, China)²

Abstract Publish/subscribe systems have recently gained significant attention. However, there is few well-defined metrics in the publish-subscribe area. In this paper, we provide a broad overview of relevant quality-of-service metrics and describe their specific meaning in the context of distributed and decentralized publish-subscribe systems. Routing decision should, in such environments, not only depend on the subscription predicate, but should also take the quality-constraints of applications and characteristics of network paths into account. This is done by extending the syntax of pub/sub system and applying four generic rulers. This paper provides a solid foundation for future work on scalable publish-subscribe services.

Keywords Quality-of-service, Metrics, Quality-constrained routing

1 引言

目前,许多分布式系统采用基于发布/订阅(P/S)的架构。发布者信息的生产者,订阅者是信息的消费者。事件是信息的载体,发布者产生事件,订阅者消费事件,他们通过订阅语言描述消费的事件类型。发布者和订阅者通常分布在网络不同的节点上,因此 P/S 系统本质上是由事件代理框架构成的网络,事件代理负责事件的匹配和路由。新型面向大规模的 P/S 系统,能够使得信息交互的双方在时间、空间和控制流三个方面都被完全解耦。高效的、大规模的事件代理框架是近年来的主要研究方向,提出了各种类型的 P/S 系统^[1-6]。但这些研究大都十分复杂,不能提供一个通用的服务质量度量,满足对系统进行比较和评价的要求。

IndiQoS 系统^[1]提供 QoS 文档,仅支持带宽、网络时延和周期或零星地交付等 QoS 度量。文献[7]中,支持持久、事务和优先权作为相关的度量。本文中把持久性归类为更一般的交付保证,并认为事务应放在事件代理框架以外去讨论。文献[8]中基于覆盖网进行分组,但维护多个独立的覆盖网的开销较大。Hermes^[9]、Choreca^[10]突破了覆盖网的多播限制,实现了基于内容的路由,未提供对网络时延、机密性 QoS 度量的支持。文献[11]中支持 P/S 中的 QoS,但它没有把 QoS 参数嵌入在事件中。本文把 QoS 参数嵌入在事件中。DDS^[12]定义了一般的中间件 QoS,但没有涉及网络 QoS、事件本身的质量和 P/S 路由等主题。另一个与本文有关的领域是消息

中间件,包括 CORBA^[13]和 JMS 等,它们处理基本的消息通信,没有足够的处理能力处理商用领域的方案。

与上述方法相比,本文的主要特点:

(1)全面研究了有关 QoS 度量,定义了它们在 P/S 系统中的语义,提出了 QoS 的三个层次;

(2)扩展了 P/S 系统的语法,使之支持带 QoS 约束的路由,并提出了事件代理框架中受限路由算法的一般原则。

2 基本概念和定义

2.1 P/S 系统中 QoS 度量

定义 1(网络时延, latency) 订阅者请求一个在最大的网络时延范围内的发布者。

端到端的网络时延取决于它们之间事件代理的跳数、每跳之间花费的时间和每个代理过滤事件的时间。在分布式环境中,这个条件不能绝对保证。一种处理方法是在发布者和订阅者之间预先分配路径,避免按请求建立连接的开销。在基于上下文的系统中,通常采用另一种更加快速的方法:对事件进行标记,并且把事件合并进通道,这避免了基于内容的过滤,简化了事件在每个代理上的路由。这种方法把基于内容的过滤映射到基于主题的过滤,但可能导致状态爆炸,最有效的是映射到网络层组播。

定义 2(带宽, bandwidth) 发布者说明所发布的事件流的上下限;订阅者限制所接收到的最大事件流。

P/S 系统总带宽取决于每一个代理的吞吐量和事件的规

^{*})本研究得到南京信息工程大学科研基金资助项目(Y640)资助。董 彪 博士生,主要研究方向为中间件技术;陈金辉 副教授,主要研究方向为系统分析与集成;孙亚民 教授,博士生导师,主要研究方向为网络工程。

模。在每一个代理上考虑带宽限制,可以计算出一条路径吞吐量的上界,从而允许基于邻居代理的最高可用带宽的事件路由算法。在 Internet 上,当事件流通过自治系统的边界时,带宽问题变得突出。在移动计算领域,P/S 系统应限制交付事件的带宽,减少订阅者的资源耗费。

定义 3(事件优先级, event priorities) 发布者说明所发送事件的相对优先级,或者和其它发布者的事件相比较的绝对优先级;订阅者说明其订阅之间的相对优先级。

在分布式事件代理框架中,事件优先级有每个代理的端到端的,它们提供绝对的、实时的优先级保证。在每一个代理过滤事件、或者向前传递事件期间,高优先数的事件将优先于低优先数的事件。在端到端的应用中,事件的优先级能用来控制每一个代理上的局部队列,并且可以使用通道缩短高优先级事件的路径。

定义 4(交付保证, delivery guarantees) 发布者说明订阅者必须接受的事件;订阅者说明哪些事件必须接受,哪些事件不太重要。

交付保证包括交付完整性和重复性。订阅者能获得至少一次,至多一次或正好一次事件的语义。如果代理框架不可靠,通过生成复制,增加交付的可能性实现至少一次的语义;采用单路径传递,或者在事件到达订阅者之前进行重复过滤实现至多一次的语义;正好一次是前两者的结合。

另一个关于交付保证的问题是考虑订阅者暂时的断开。在移动环境中,事件代理必须存储事件,直到订阅者变得再一次可用,以保证事件至少被传递一次。断开时间可以是任意长的时间,代理框架将在一个有意义的时间间隔内存储事件。

定义 5(订阅的选择性, selectivity of subscriptions) 订阅者用某种语言定义它们的订阅条件。

能用不同类型的语言表达订阅。简单订阅包括一个主题标识,复杂订阅支持属性匹配。根据过滤特点、分布式匹配算法和交付的错误率等,订阅语言有一个关于复杂性的层次划分,复杂语言通过精确的订阅能减少交付的错误率,但是,复杂语言使得过滤、优化等更加困难。

在整个框架中,订阅语言可以不同。一个代理可从它的订阅者接收一个复杂订阅,而向邻居代理传递一个简单版本的订阅,该版本能被它的邻居代理明白,并且更适合拓扑结构。

定义 6(周期或零星交付, periodic or sporadic delivery) 发布者周期地或零星发布事件;订阅者周期地或零星地接收订阅。

周期性发布的事件代表了与该次发布同时的发布者状态。零星发布的事件仅仅当发布的事件发生变化时才被发布,可作为一个预先过滤事件,当事件的变化超出某一门限后才发布。

定义 7(事件顺序, event order) 发布者说明发布事件的次序;订阅者说明接受匹配事件的次序。

在很多情况下,可采用集中的顺序、有序的传输(ATM、TCP)或者让发布者(或者代理)标识一个顺序等,满足顺序条件。但是,事件到达的顺序可能相关,也可能不相关。

在事件交付期间,代理框架可能改变事件的顺序,从而影响事件到达订阅者的顺序,甚至引起事件的丢失。事件顺序交付不包含有保证的交付。一个有效方案是如果一个事件的后继事件已经被交付,就丢弃该事件。如果不允许丢弃该事件,该事件必须在发送前记录下来,这可能引起任意长的网络时延。

分布式事件排序是另一个问题,对于基于内容的订阅,这

是困难的。定义一个有意义的顺序,在相当程度上依靠订阅语言和应用。一般的途径是一个通过中心代理强制标识一个全局顺序,这限制了系统的大规模性。

定义 8(有效间隔, validity interval) 发布者说明事件超时,或者后继事件表示与前一事件的无关性。

对于框架来说,知道一个事件在多长时间有效是重要的。有效间隔既可用时间来标识,又可通过后继事件来推断,基于跳计数的时间有效性在 P/S 系统中是无意义的。如果只对最近的事件感兴趣,通过后继事件标识有效间隔是一种有效的途径。

定义 9(数据源冗余, source redundancy) 对于同一事件,订阅者要求冗余的数据源。

数据源冗余增强了框架的容错能力,允许订阅者核查事件,可作为交付保证的实现途径。冗余性要求发布者的事件是可比较的,复杂的订阅语言可能阻碍数据源冗余。

定义 10(机密性, confidentiality) 订阅者加密它们的订阅,或者把订阅发送到可信代理;发布者只连接到可信代理,或者发送加密事件。

机密性在可信代理的子网中是容易实现的。在不可信网络的交付路径上,可信代理之间使用标准的加密机制,并且强迫不可信代理把事件广播到它的所有邻居,从而实现机密性。这时,不可信代理具有较高的负载,也引起大量的重复事件的传播。

在基于主题的 P/S 系统中,在不可信的代理网络上,事件的内容对匹配来说是透明的,事件能被加密。而在不可信代理参与匹配处理时,机密性的实现变得困难,如基于内容的匹配要求比较深入地了解事件的内容。

大量数据库领域的文献提供了不解密查询加密数据的方法。在 P/S 系统能借鉴这些方法,可在不可信代理上实现盲匹配。但是,在任意的一种查询和数据之间,实现盲匹配是不现实的。

定义 11(认证, Authentication) 包括发布者和订阅者认证。

发布者采取存取控制,通过数字水印保证事件的完整性。订阅者使用数字签名认证订阅。

2.2 QoS 的层次模型

QoS 的层次模型分为网络层 QoS、事件代理框架层 QoS 和应用层 QoS。

定义 12(网络层 QoS) 提供系统配置以及连接 P/S 系统的发布者、订阅者和事件代理框架,使用度量函数获取的观察值,包括网络时延、带宽、事件优先级和交付保证。该层是模型的下层,记为 QoS_{net} 。

定义 13(事件代理框架层 QoS) 为一个特性集,控制 P/S 系统的行为,由发布者 QoS 和订阅者 QoS 两类组成,服从“发布者提供-订阅者请求”模式。在这种模式中,订阅者说明所要求的 QoS 的值,发布者说明所提供的 QoS 的值,当且仅当发布者的 QoS 值与订阅者的 QoS 值兼容时,框架才在两者之间建立联系。包括订阅的选择性、周期或零星交付、事件顺序、有效间隔、数据源冗余、机密性和认证。该层是模型的中层,记为 QoS_{middle} 。

定义 14(应用层 QoS) 指嵌入在事件中的信息质量,包括数据项和信息声明的完整性、正确性、时效性、一致性和精确性。P/S 系统分析事件的内容,然后按照一些预先定义的准则估值信息质量,这个分析过程与领域知识相关。该层是模型的上层,记为 QoS_{app} 。

3 带 QoS 约束的路由

3.1 P/S 系统操作的扩展

定义 15 P/S 系统的扩展操作由 register(S, pt, Q)、cancel(S, pt, Q)、pub(P, e, Q) 和 notify(S, e, Q) 四种操作组成。S 为订阅者, P 为发布者, pt 为订阅条件, $Q = \{ QoS_{net}, QoS_{middle}, QoS_{app} \}$, e 为发布的事件。其中:

- (1) register(S, pt, Q): S 订阅 pt, 并且对三层 QoS 有要求。
- (2) cancel(S, pt, Q): S 取消订阅 pt。
- (3) pub(P, e, Q): P 发布 e, 并且提供三层 QoS。
- (4) notify(S, e, Q): e 与 S 的订阅 pt 匹配, 并且 e 提供的 QoS 满足 S 中相应 QoS 的要求。

扩展发布/订阅操作中 QoS_{net} , QoS_{middle} , QoS_{app} 没有绑定发布者和订阅者, 维持了发布者和订阅者之间的解耦性。

3.2 带 QoS 约束的路由规则

带 QoS 约束的路由是带质量约束的路由, 表示路径决策满足 QoS 要求的路由。

定义 16 “交付保证”规则 $Rule_{gd}$:

register(S, pt, Q) \Rightarrow notify(S, e, Q)

如果 S 订阅 pt, P/S 系统把 e 告知 S, 这意味着 P/S 系统保证把一个事件交付到所有已经订阅它的用户。

定义 17 “发布者提供-订阅者申请”模式规则 $Rule_{ps}$:

satisfy(register(S, pt, Q), pub(S, e, Q)) \Rightarrow notify(S, e, Q)

其中: satisfy(sub, pub) 是模式估值函数, 它的参数是 e 的发布者和订阅者。

$Rule_{ps}$ 表示当“订阅者申请-发布者提供”模式的得到满足, e 被交付给 S, S 已经订阅 pt, 并且 pt 与 e 匹配。

定义 18 “发布者按信息质量选择发布”规则 $Rule_{app}$:

quality(pub(P_1 , e_1 , Q_1), pub(P_2 , e_2 , Q_2)) \Rightarrow notify(S, e_1 , Q_1)

其中: quality 用于比较两个发布者所发布数据的数据质量, 参数是 e 的两个发布者 P_1, P_2 。

当 e 有一个以上的发布者时, $Rule_{app}$ 才起作用。代理框架选择一个发布者 P_1 , 并且阻止其它发布者的事件 e, 然后向前传递 P_1 发布的事件 e。例如, 两个发布者跟踪两个不同的传感器数据, 代理框架选择有较好精度的事件进行传递。

定义 19 “发布者按网络层 QoS 选择发布”规则 $Rule_{net}$:

cost(pub(P_1 , e_1 , Q), register(S, pt, Q)) > cost(pub(P_2 , e_2 , Q), register(S, pt, Q)) \Rightarrow notify(S, e_2 , Q)

其中: cost(pub, register) 估值事件的发布开销, 参数是事件的发布者和订阅者。

与 $Rule_{ps}$ 一样, 当事件有一个以上的发布者时, $Rule_{net}$ 才起作用。代理框架根据网络 QoS, 比较两对发布者、订阅者之间的事件交付开销, 选择开销小的发布者发布 e。

算法 1 带 QoS 约束的路由

QoSCon_Routing(S, P, e, Q)

{ App_Rule_{gd}; //应用 Rule_{gd}

App_Rule_{ps}; //应用 Rule_{ps}

if 有两个以上的发布者

{ App_Rule_{app}; //应用 Rule_{app}

App_Rule_{net}; } //应用 Rule_{net}

路由事件 e;

由于事件代理框架中 QoS 是高度动态的, 规则的实现要考虑到 QoS 质量的震荡。例如: 随着工作负载的激增, 低负载的链路将迅速过载。

结束语 本文系统地定义了大规模 P/S 系统中 QoS 度量的语义, 为事件代理框架提供了表达 QoS 请求的方法, 标识了不同层次的质量水平, 通过扩展 P/S 操作, 提出了带 QoS 约束条件的路由原则。依据本文的研究成果, 各种 P/S 系统能被比较和整合到一个前沿领域, 为未来大规模 P/S 系统提供坚实的基础。

参考文献

- [1] Carvalho N, Araújo F, Rodrigues L. Scalable QoS-based Event Routing in Publish-Subscribe Systems. NCA, 2005; 101-108
- [2] Rebeca A. Event-based Electronic Commerce Architecture. <http://www.gkec.informatik.tu-darmstadt.de/rebeca/> (accessed August 2007)
- [3] Terpstra W W, Behnel S, Fiege L, et al. Bit Zipper Rendezvous-Optimal Data Placement for General P2 PQueries // Proc. of the 1st Int. Workshop on Peer-to-peer Computing and Databases, Heraklion, Crete, Mar. 2004; 466-475
- [4] Belokosztolszki A, Eyers D M, Pietzuch P R, et al. Role-based access control for publish/subscribe middleware architectures // DEBS03, New York, 2003
- [5] Rowstron A, Druschel P. Pastry: Scalable, decentralized object location, and routing for large-scale Peer-to-Peer systems // Proc. of the Int. Middleware Conference (Middleware2001). Nov. 2001; 329-350
- [6] Rowstron A, Kermarrec A M, Castro M, et al. SCRIBE: The design of a large-scale event notification infrastructure // Crowcroft J, Hofmann M, eds. Proc. of the 3rd Int. Workshop on Networked Group Communications (NGC'01), London, UK, 2001; 30-43
- [7] Eugster P T, Felber P A, Guerraoui R, et al. The many faces of publish/subscribe. ACM Computing Surveys, 2003, 35(2): 114-131
- [8] Castro M, Jones M B, Kermarrec A M, et al. An evaluation of scalable application-level multicast using peer-to-peer overlays // INFOCOM. 2003; 1510-1520
- [9] Pietzuch P, Bacon J. Hermes: A distributed event-based middleware architecture // Proc. of the 1st Int. Workshop on Distributed Eventbased Systems (DEBS'02). Vienna, Austria, 2002; 611-618
- [10] Terpstra W W, Behnel S, Fiege L, et al. A Peer-to-Peer Approach to Content-based Publish/Subscribe // DEBS03, San Diego, CA, 2003; 1-8
- [11] Araujo F, Rodrigues L. Quality of Service in Indirect Communication Systems // Fourth European Research Seminar on Advances in Distributed Systems (ERSADS'01)
- [12] Data Distribution Service for Real-Time Systems Specification, ptc/06-04-11. http://www.omg.org/technology/documents/formal/data_distribution.htm (accessed August 2007)
- [13] Object Management Group. CORBA notification service, version 1.0.1. OMG Document formal/2002-08-04, 2002