

文章编号:1001-9081(2009)05-1225-05

## 发布/订阅范型中信息投递的服务质量保证

袁 媛

(重庆邮电大学 计算机科学与技术学院, 重庆 400065)

(yuanyuan@cqupt.edu.cn)

**摘 要:**发布/订阅范型中信息可靠投递是保证系统服务质量的关键,现有研究对此考虑不充分。以发布/订阅范型在时间维度上解耦导致的信息投递不确定性为出发点,采用时序分析方法,研究信息投递服务质量的保障措施。在定义投递信息可靠性的基础上,探讨了订阅活跃期、订阅延迟、发布传播延迟对信息投递服务质量的影响,给出了保证服务质量的分布式通知服务的活性性质,并设计了一种提供服务质量保证的发布/订阅代理。分析结果表明,文中给出的计算性质是保障信息投递服务质量的必要条件,信息投递服务质量可由可靠的分布式通知服务得到保证。

**关键词:**发布/订阅范型;服务质量;分布式通知服务;信息投递;事件

**中图分类号:** TP338.8 **文献标志码:** A

## QoS guarantee of the information delivery in publish/subscribe paradigm

YUAN Yuan

(Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)

**Abstract:** Reliable information delivery to subscriber in publish/subscribe paradigm is the key to Quality of Service (QoS) guarantee of the system. However current researchers have not resolved this problem adequately. The author took the non-determinism introduced by decoupling on time dimension as starting point. In temporal analysis method, the principles of QoS guarantee for the information delivery in the paradigm were surveyed. After defining the reliability of delivery information, the author concentrated on the effect of the active interval of subscription, and the delay of subscription, and the delay of publication diffusion on QoS of the information delivery. Then the author provided liveness property of distributed notification service for ensuring the QoS, and devised a publish/subscribe broker to ensure the QoS. The result of analysis shows that the computational property provided here is vital to guarantee QoS of the information delivery, and QoS of the information delivery can be ensured by reliable distributed notification service.

**Key words:** publish/subscribe paradigm; Quality of Service (QoS); Distributed Notification Service (DNS); information delivery; event

### 0 引言

发布/订阅 (Publish/Subscribe, Pub/Sub) 范型是广域范围内分布式系统信息集散的有效方法<sup>[1-2]</sup>, 订阅者和发布者两个进程角色通过提供信息共享的分布式通知服务 (Distributed Notification Service, DNS) 互换信息, 发布者向分布式通知服务发放发布信息 (Publication), 订阅者向分布式通知服务提交订阅 (Subscription) 以表示对特定信息的兴趣, 如果订阅与发布信息匹配, 分布式通知服务向订阅者发放一个承载发布信息的通知 (Notification)。与传统基于 RPC 的信息交换模式相比, 其突出优点体现在: 分布式构件通过分布式通知服务进行间接通信, 不需要对它们实施绑定; 交互活动参与方彼此独立, 无须明确知道对方的存在; 通信双方不必同时处于运行态, 出现通信不畅时不需要阻塞参与者。因此, 它为分布式构件提供了一种多对多的异步通信模式, 参与方在空间、时间和信息流三个方向上的耦合被完全解除。基于上述优点, Pub/Sub 范型尤其适合大规模、动态变化分布式应用的信息共享<sup>[3-4]</sup>, 现已应用在金融<sup>[5]</sup>、股票<sup>[6]</sup>、工作流<sup>[7]</sup>、网络游戏<sup>[8]</sup>、移动计算<sup>[9]</sup>等场合, 而 SIENA<sup>[10]</sup>、Scribe<sup>[11]</sup>、A-MEDIAS<sup>[12]</sup>等是一些较有影响的 Pub/Sub 原型。

软件工程领域虽已从可伸缩性、Pub/Sub 范型、路由机制

和匹配机制等方面对基于 Pub/Sub 范型的分布式系统做了大量的研究<sup>[13-14]</sup>, 但为了系统能提供可信的服务, 必须尽量避免不确定行为的发生, 这是保障资源持久共享的关键<sup>[15]</sup>。Pub/Sub 范型的不确定性首先表现为进程失效或崩溃, 并发生进程导致的不可预测性、通信故障等典型分布式系统的不确定问题。进而, 由于在时间、空间、控制流维度上的耦合被解除, 系统可扩展性特别良好, 但却降低了系统可靠性, 这集中体现为时间维解耦导致的两种不确定情形: 1) 发布者可能在订阅处于非激活态发布信息, 此时订阅者不会对任何信息感兴趣; 2) 发布者可能在订阅已经“过期”时发布信息, 此时订阅者已不再对特定信息感兴趣。在出现上述情况时, 系统不应将信息强行投递给订阅者, 否则将导致错误的信息投递。这种信息投递的不确定性是基于发布/订阅范型系统特有的, 该问题被称为信息的可靠投递问题, 它是保证基于 Pub/Sub 范型系统服务质量 (Quality of Service, QoS) 的前提<sup>[16]</sup>。

在基于主题和基于内容两类主要 Pub/Sub 范型的研究领域内, 侧重从安全性和事务研究系统的 QoS, 主要通过约束发布信息和订阅来保证系统 QoS。文献[17]给出了 Pub/Sub 系统认证、机密性等安全需求, 并使用 Cache 机制暂存信息内容达到提高 QoS 的目的; 文献[18]假定存在一个可信任的代理网络, 用 SIENA 的覆盖关系来约束可行的订阅和广告; 文献[19]把基于角色的访问控制和分布式通知服务结合起来, 并

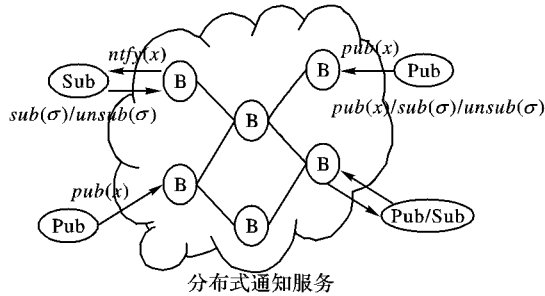
收稿日期: 2008-11-12; 修回日期: 2009-01-18。

作者简介: 袁媛 (1972-), 男, 四川自贡人, 讲师, 博士, CCF 会员, 主要研究方向: 分布式计算、Web 服务。

给出了一种基于 Hermes 的扩展原型系统;文献[20]则采用基于时间回应的事务机制来保证信息的 QoS,但这种方法的实质是在订阅者得不到感兴趣信息时的一种重新订阅机制。上述方法的共同点是:仅从安全性和事务提高系统 QoS,而没有从分布式组件可预期性的角度研究系统 QoS,而这一点是保证系统可信的关键,上述工作在此方面却未曾涉足。为得到可信的 Pub/Sub 服务,本文围绕 Pub/Sub 分布式计算范型中信息投递的 QoS 展开讨论。下文在分析投递信息可靠性的基础上,给出了保证信息投递 QoS 的一些基本性质,通过对服务通知活性条件的讨论,分析了分布式通知服务保证信息投递 QoS 的具体措施。

## 1 计算模型

考虑由进程集合  $P = \{p_1, \dots, p_n\}$  构成的分布式系统,系统存在一个离散的全局时钟,时钟取值范围为自然数集合。进程在同一时刻可同时扮演两种角色:信息产生者(订阅者)或信息消耗者(发布者),并通过分布式通知服务间接通信而交换信息,分布式通知服务是一个由发布/订阅 Broker  $B = \{B_1, B_2, \dots, B_r\}$  通过覆盖网络连接的信息传输介质,发布/订阅 Broker 是发布信息和订阅的代理,以下简称 Broker。本文 Pub/Sub 的计算模型如图 1 所示。



注: Pub: 发布者进程; Sub: 订阅者进程;  
Pub/Sub: 扮演两种角色的进程; B: 发布/订阅 Broker 进程。

图 1 基于分布式通知服务的 Pub/Sub 计算模型

$t$  时刻跨越 DNS 边界的操作包括:进程  $p \in P$  执行的操作  $pub(x)$ 、 $sub(\sigma)$ 、 $unsub(\sigma)$ , 分别表示发布者向 Broker 发布信息  $x$ 、订阅者向 Broker 注册订阅  $\sigma$  和从 Broker 中取消订阅  $\sigma$ ; Broker 执行  $ntfy(x)$  操作,向订阅者发放通知,订阅者则得到投递信息  $x$ 。订阅者、发布者或 Broker 在  $t$  时刻每执行一个操作,  $p_i \in P$  的状态都将发生一次变迁,并在  $p_i$  上发生一个事件  $e_i(op, t)$ , 其中,  $t$  为事件发生时间,  $op$  为操作集合  $\{pub(x), sub(\sigma), unsub(\sigma), ntfy(x)\}$ , 进程  $p_i \in P$  执行  $pub(x)$ 、 $sub(\sigma)$ 、 $unsub(\sigma)$  操作分别发生事件  $e_i(pub(x), tp)$ 、 $e_i(sub(\sigma), ts)$  和  $e_i(unsub(\sigma), tu)$ , 而 Broker 进程执行  $ntfy(x)$  操作将在  $p_i$  上发生事件  $e_i(ntfy(x), tn)$ 。订阅  $\sigma = (\varphi, p, t)$ , 其中, 订阅条件  $\varphi$  是由约束(属性, 操作符, 值)的合取构成的谓词表达式,  $t$  为订阅者  $p$  提交  $\sigma$  的时间戳, 发布信息  $x$  是(属性, 值)对的集合,  $\sigma$  与  $x$  匹配表示为  $x \sqsubset \sigma$ 。

发布信息  $x$  一经在代理  $x$  的 Broker 上产生将在 Broker 网络内传播,被  $x$  覆盖的 Broker 将  $x$  与注册在其上的  $\sigma$  匹配,如果匹配成功, Broker 执行  $ntfy(x)$  向对  $x$  感兴趣的订阅者投递  $x$ , 否则该 Broker 将  $x$  传播给相邻 Broker,若订阅者在可容忍的时段内还未得到感兴趣的  $x$ , 将执行  $unsub(\sigma)$  操作从 Broker 中取消订阅。为便于讨论,同一信息  $x$  在与  $\sigma$  成功匹配之前称为发布或发布信息,得到匹配后称为通知或投递信息,且为保证 DNS 中资源的唯一标示,认为在  $t$  时刻,两个  $pub(x)$  操作不会发放相同信息  $x$ , 订阅者进程发放至多一个订阅  $\sigma$ 。由于 Broker 的匹配算法、发布信息和订阅的具体数

据结构及确定代理它们 Broker 的方法与本文讨论无关,下文的讨论中认为这些因素已给出。

## 2 信息投递的可靠性分析

考虑发布者在  $tp$  时刻发布信息  $x$ , 理论上,在  $t > tp$  时刻所有对  $x$  感兴趣的订阅者都认为  $x$  是可靠信息,但由于 Pub/Sub 范型在时间上的松耦合特性,订阅者可能未提交任何订阅,也有可能已对  $x$  不再感兴趣,  $x$  不能被随便地投递到订阅者上。具体而言,以布尔谓词  $AV_i(x, \sigma, t)$  表示信息  $x$  在  $t$  时刻上对订阅者  $p_i \in P$  发放的订阅  $\sigma$  的可靠性,订阅者在  $t$  时刻可能存在如下三种状态:

- 1)  $p$  未发放订阅  $\sigma$ , 订阅是非活跃的,  $AV_i(x, p, t) \equiv \perp$ ;
- 2)  $p$  已在  $ts < t$  时刻向 DNS 提交订阅  $\sigma$ , 订阅是活跃的,  $AV_i(x, p, t) \equiv \top$ ;
- 3)  $p$  在提交订阅  $\sigma$  的  $\Delta t$  时段后对  $x$  不再感兴趣,则取消订阅  $\sigma$ ,  $AV_i(x, \sigma, t) \equiv \perp$ ;

将上述  $\Delta t$  表示为  $I_i(\sigma)$ , 它等于两个连续事件  $e_i(sub(\sigma), ts)$ 、 $e_i(unsub(\sigma), tu)$  ( $ts < tu$ ) 发生时刻的差值  $tu - ts$ , 而  $[ts, ts + I_i(\sigma)]$  称为订阅者  $p_i$  订阅  $\sigma$  的活跃时段。只有发生在订阅者  $p_i$  订阅  $\sigma$  的活跃时段内的通知事件给出的投递信息才是可靠的,其语义为订阅者不能得到它不感兴趣的通知。因此,确保通知事件发生在订阅的活跃时段内是保障信息投递 QoS 的关键。

如图 2 所示是一个信息可靠投递实例。图中包含四个进程,  $p_i$ 、 $p_j$  为订阅者,  $p_k$  为发布者,  $p_j$  扮演了订阅者和发布者两种角色。 $p_i$  在  $ts_1$  发放订阅  $\sigma_1$ , 在  $tu_1$  取消订阅  $\sigma_1$ ,  $I_i(\sigma_1)$  为  $p_i$  的  $\sigma_1$  的活跃时段, 在  $ts_2$  发放订阅  $\sigma_2$ , 在  $tu_2$  取消订阅  $\sigma_2$ ,  $I_i(\sigma_2)$  为  $p_i$  的  $\sigma_2$  的活跃时段;  $p_j$  发放订阅  $\sigma_1$  和  $\sigma_2$  的活跃时段分别为  $I_j(\sigma_1)$  和  $I_j(\sigma_2)$ , 并在  $tp_2$  作为发布者发布信息  $y$ ;  $p_k$  发放订阅  $\sigma_1$  和  $\sigma_2$  的活跃时段分别为  $I_k(\sigma_1)$  和  $I_k(\sigma_2)$ ;  $p_h$  在  $tp_1$  时刻发布信息  $x$ 。

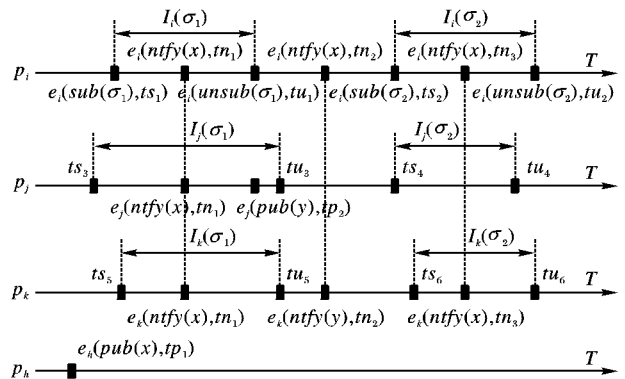


图 2 信息可靠投递实例

信息  $x$  在  $p_i$  上产生通知的时刻分别为  $tn_1$ 、 $tn_3$ , 信息  $y$  在  $p_i$  上产生通知的时刻为  $tn_2$ ; 信息  $x$  在  $p_j$  上产生通知的时刻为  $tn_1$ ; 信息  $x$  在  $p_k$  上产生通知的时刻分别为  $tn_1$ 、 $tn_3$ , 信息  $y$  在  $p_k$  上产生通知的时刻为  $tn_2$ 。根据上文讨论, 由于  $tn_1 \in [ts_1, ts_1 + I_i(\sigma_1)]$ ,  $tn_3 \in [ts_2, ts_2 + I_i(\sigma_2)]$ , 因此  $p_i$  在  $tn_1$ 、 $tn_3$  得到的投递信息  $x$  是可靠的, 但  $tn_2 \notin [ts_1, ts_1 + I_i(\sigma_1)] \vee [ts_2, ts_2 + I_i(\sigma_2)]$ , 因此  $p_i$  在  $tn_2$  得到的投递信息  $y$  是不可靠的; 同理,  $p_j$  在  $tn_1$  得到的投递信息  $x$  是可靠的; 而  $p_k$  在  $tn_1$ 、 $tn_4$  得到的投递信息  $x$  是可靠的, 在  $tn_3$  得到的投递信息  $y$  是不可靠的。

## 3 分布式通知服务对信息投递 QoS 的保证

由上文的讨论可知,信息的可靠投递主要是要保证通知事件在订阅活跃时段内发生,理想情况下,订阅活跃时段是从

订阅者发放订阅到取消订阅的时间区间,但实际中它受各种环境因素的制约,主要体现为 DNS 对信息投递可靠性的影响,Broker 结构和信息处理能力、DNS 的网络性能等因素直接影响着投递信息 QoS。

DNS 的基本任务包括两方面:1) 存储并管理订阅者的订阅  $\sigma$ , 这与 Broker 本身的物理结构和性能有关;2) 发布者发布信息  $x$  后,代理  $x$  的 Broker 将  $x$  与  $\sigma$  匹配,若成功匹配则将  $x$  通知到感兴趣的订阅者,否则按路由协议在 Broker 网络中传播  $x$ 。如果不考虑进程与 DNS 之间的通信延迟,即  $pub(x)$ 、 $sub(\sigma)$ 、 $unsub(\sigma)$ 、 $ntfy(x)$  四个操作引起的通信延迟为 0, DNS 边界内通信系统对信息投递 QoS 的影响包括两方面通信延迟:1) 从 DNS 处理订阅者订阅  $\sigma$  角度考虑,进程将  $\sigma$  注册到 Broker 或从 Broker 取消  $\sigma$  导致的订阅延迟;2) 从 DNS 处理发布者发布信息  $x$  角度考虑,  $x$  在 Broker 间的传播延迟。

### 3.1 订阅延迟对信息可靠投递的影响

当订阅者向 DNS 发放订阅  $\sigma$  时,代理  $\sigma$  的 Broker 将开销一定的时间将  $\sigma$  注册到订阅表中;而订阅者从 DNS 中取消  $\sigma$  时,NS 将开销一定的时间将订阅从订阅表中删除。因此从进程上发生订阅或取消订阅事件到 Broker 感知订阅表的变化将有一定延迟,这种延迟主要由 Broker 的结构和信息处理能力决定,主要与 Broker 结构组织的复杂性、订阅表的存储结构和访问速度有关。采用  $D_{sub}(\sigma)$  表示从订阅者  $p$  执行  $sub(\sigma)$  操作到订阅  $\sigma$  注册到 Broker 的延迟,如果  $D_{sub}(\sigma) > I(\sigma)$ , 说明 Broker 还来不及注册  $\sigma$ ,发放  $\sigma$  的订阅者就已经不再对发布信息感兴趣,订阅者永远不能得到感兴趣的发布信息;否则  $D_{sub}(\sigma) < I(\sigma)$ , 称  $p$  的订阅  $\sigma$  在  $[ts + D_{sub}(\sigma), ts + I(\sigma)]$  时段上是稳定的,称这一时段为  $\sigma$  的稳定时段,这时信息必须在被订阅者发放的订阅稳定后才能投递,由于  $[ts + D_{sub}(\sigma), ts + I(\sigma)] \subset [ts, ts + I(\sigma)]$ , 因此在考虑订阅延迟时,信息投递的可靠性将降低。如图 3 所示为分布式通知服务延迟对信息可靠投递的影响。考虑  $p_k$ 、 $p_h$  上的订阅  $\sigma_1$ 、 $\sigma_2$ ,  $\sigma_1$  的稳定订阅时段为  $[ts_1 + D_{sub}(\sigma_1), ts_1 + I(\sigma_1)]$ ,  $\sigma_2$  的稳定订阅时段为  $[ts_2 + D_{sub}(\sigma_2), ts_2 + I(\sigma_2)]$ , 如果不考虑订阅延迟,  $e_k(ntfy(x), tn_2)$  的发生将使  $p_h$  得到投递信息  $x$ ;但在考虑订阅延迟时,由于事件没有发生在  $\sigma_1$  的稳定时段内,因此  $p_h$  不能得到  $x$ , 信息投递 QoS 得不到保证。

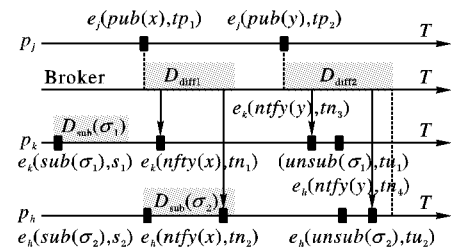


图3 分布式通知服务延迟对信息可靠投递的影响

### 3.2 发布传播延迟对信息可靠投递的影响

传播延迟主要考虑发布信息在 Broker 间广播转发的网络延迟。发布者  $p_i$  在  $tp$  时刻发布信息  $x$ , 且订阅者  $p_j$  在时刻  $tp$  上存在稳定匹配订阅  $\sigma$ , 即  $\sigma.t \in [ts + D_{sub}(\sigma), ts + I(\sigma)]$ 。为了向  $p_j$  发放投递信息  $x$ ,  $x$  将在 DNS 的 Broker 间扩散,  $\Delta i$  为信息  $x$  扩散到可能的 Broker  $B_k \in \{B_1, \dots, B_m\}$  的时间, 事件  $e_j(ntfy(x), tn)$  只有在  $tn \in [tp, tp + \Delta i]$  时刻发生,  $p_j$  才能得到投递信息  $x$ 。考虑最极端的情况, 订阅者进程  $\{p_1, p_2, \dots, p_n\}$  都对  $x$  感兴趣, 此时扩散延迟将取  $\{\Delta_1, \dots, \Delta_n\}$  中的最大值, 即  $D_{diff} = \max \{\Delta_1, \dots, \Delta_n\}$ , 这表示通知事件不应该发生在最大传播延迟之外, 即 DNS 必须在  $[tp, tp + D_{diff}]$  时段内向对  $x$  感兴趣的订阅者发放投递信息, 或者说所有感兴趣的

订阅者  $p$  应该在这一时段内得到通知。如图 3 所示, 进程  $p_k$  在  $tp_2$  时刻发布  $y$ ,  $y$  在 Broker 网络中的最大传播延迟为  $D_{diff}$ , 若  $y$  传播到图中 Broker 时与  $p_h$  订阅  $\sigma_2$  成功匹配, 则该 Broker 在  $tn_4$  时刻向  $p_h$  投递信息  $y$ , 但此时  $p_h$  已经取消了订阅  $\sigma_2$ 。因此如果网络性能不好, 将导致信息的传播延迟过大而出现不可靠的信息投递, 也不能保证信息投递 QoS。

### 3.3 保证信息投递服务质量的活性性质

在考虑 DNS 两类延迟  $D_{sub}$ 、 $D_{diff}$  的情况下, 为保证信息的可靠投递, 通知事件的发生时间将受到一定约束, 这时通知事件应该发生在订阅的稳定时段内, 且不能超出发布信息在 DNS 中传播的最大时间延迟。因此有如下保证信息投递 QoS 的活性性质:

$$\forall i \exists t'(e_i(pub(x), t') \wedge ([t', t' + D_{diff}] \subset [\sigma.t + D_{sub}, \sigma.t + I(\sigma)])) \Rightarrow \exists t''(e_{\sigma,p}(ntfy(x), t'') \in H) \wedge t'' \in [t', t' + D_{diff}])$$

这一性质表明: 投递信息必须在传播延迟内投递到订阅稳定的订阅者才能保证信息投递 QoS。

图 3 中, 如果  $p_k$  的订阅  $\sigma_1$  分别在  $tn_1$  和  $tn_3$  时刻与发布信息  $x$  和  $y$  得到匹配, 由于  $tn_1, tn_3 \in [ts_1 + D_{sub}, ts_1 + I(\sigma_1)]$ , 且  $tn_1 \in [tp_1, tp_1 + D_{diff}]$ 、 $tn_3 \in [tp_2, tp_2 + D_{diff}]$ , 因此在  $x$  和  $y$  的扩散延迟  $D_{diff}$  和  $D_{diff}$  内  $\sigma_1$  是稳定的。由活性性质描述可知,  $e_k(ntfy(x), tn_1)$  和  $e_k(ntfy(y), tn_3)$  是两个可靠通知事件, 信息  $x$  和  $y$  向  $p_k$  的投递是可靠的, 而考虑在  $p_h$  上发生的通知事件  $e_h(ntfy(x), tn_2)$ , 虽然  $tn_2 \in [tp_1, tp_1 + D_{diff}]$ , 但  $tn_2 \notin [ts_2 + D_{sub}, ts_2 + I(\sigma)]$ , 这说明 Broker 向  $p_h$  发放通知  $x$  时  $\sigma_2$  还未稳定, 因此  $e_h(ntfy(x), tn_2)$  不是可靠通知事件, 信息  $x$  在  $tn_2$  时刻向  $p_h$  的投递是不可靠的, 但在不考虑或减小订阅延迟的情况下可保证信息  $x$  在  $tn_2$  时刻向  $p_h$  的投递 QoS。如果要保证信息投递 QoS, 还必须满足订阅者在取消订阅前肯定能得到通知, 即  $tp + D_{diff} < ts + I(\sigma)$ , 这说明信息应该在订阅者取消订阅前扩散到 DNS 的所有 Broker 中。在  $ts, tp, I(\sigma)$  一定的情况下, 如果 Broker 性能越好,  $D_{sub}$  将越小, 订阅的稳定期越长, 订阅者得到与  $\sigma$  匹配的投递信息的可能则越大, 也越容易保证信息投递的 QoS; 否则  $[ts + D_{sub}, ts + I(\sigma)]$  时段将越小, 甚至可能出现  $D_{sub} > I(\sigma)$ ,  $\sigma$  将永远得不到感兴趣的投递信息。如果网络性能越好, 信息在 DNS 中的扩散延迟便越小, 则更不容易出现订阅者已取消  $\sigma$  而  $x$  还未扩散到所有 Broker 的情况, 订阅者得到与  $\sigma$  匹配的投递信息的可能则越大, 信息投递的 QoS 越容易得到保证; 否则可能出现  $tp + D_{diff} > ts + I(\sigma)$  的情况, 活性条件不成立, 不能确保订阅者一定能得到投递信息(图 3 中  $e_k(ntfy(y), tn_4)$ ), 但由于  $D_{diff}$  取的是网络最大扩散时延, 订阅者也有可能得到  $x$  的通知(图 3 中  $e_k(ntfy(y), tn_3)$ ), 因此, 此时信息投递 QoS 的保证是不确定的。

通过上述讨论可得出如下结论: 1) DNS 的活性性质是保证信息投递 QoS 的必要条件, 如果 DNS 中的通知事件满足活性性质, 发布信息  $x$  肯定能送达对其感兴趣的订阅者, 否则 DNS 不能提供足够的 QoS 保证; 2) 订阅活跃时段越长、订阅延迟越小、发布传播延迟越小, 越容易保证信息投递 QoS; 3) 与订阅延迟和发布传播延迟相关的 DNS 不确定性直接影响到信息投递 QoS, 这主要包括 Broker 结构及信息处理能力、网络传输协议和吞吐量、信息发布点、订阅发放点等因素。

## 4 发布/订阅代理对信息投递 QoS 的保证

设分布式系统中的进程按照图 1 计算模型异步地交换信息, 如果进程不会失效且操作时间有限, 信道可靠, 信息传输

延迟有限,考虑 DNS 不确定因素对信息投递 QoS 的保证。由于篇幅受限,本文暂不考虑网络传输协议和吞吐量、信息发布点、订阅发放点等因素,以 Broker 的信息处理能力和结构设计示范 DNS 对信息投递 QoS 的保证,以下讨论认为  $x$  可能覆盖到 DNS 中所有 Broker。

#### 4.1 持久信息发布的保证

如果 DNS 状况恶劣,通常不能很好地保证信息投递 QoS,但如果 Broker 具有一定信息保持能力,Broker 在向订阅者发放投递信息  $x$  后不会立即将其删除,而是在其内存中将  $x$  存储一定时段,如果需要 Broker 在存储时段内的某个时刻代替发布者重新发布  $x$ ,这是一种增强信息可靠投递的有效方法。根据 Broker 的信息保持能力,有三类 Broker 实现风格。

0-保留:Broker 无须存储投递信息  $x$ , $x$  在订阅者得到  $x$  时即刻作废。虽然不能很好地保证信息的可靠投递,但订阅和发布之间的延迟较小。这种风格在上文的讨论中已给出,它通常用于股票交易等有实时性要求的系统中。

$\Delta$ -保留:Broker 在一个确定时段  $\Delta > 0$  内存存储投递信息  $x$ , $x$  在订阅者得到  $x$  的  $\Delta$  时段后作废。信息的保留可增大其可靠投递的可能性,但必须开销一定的内存,并需要对已作废投递信息  $x$  实施垃圾回收。个性化新闻发布是这种实现风格的典型应用。

$\infty$ -保留:Broker 在应用认为足够长的时段内存存储投递信息  $x$ , $x$  不会作废。信息一旦被发布将永久存在于 Broker 中,并无须回收垃圾,当订阅者进程发放新订阅  $\sigma$  后,可以接收到所有先前发布到 DNS 中与  $\sigma$  匹配的投递信息,因此这是一种完全可靠的实现风格,但海量内存的存储代价极高。典型应用是数据库系统永久性存储目录的信息发布。

设  $\Delta_i(x)$  为发布者  $p_i$  发布的信息  $x$  被通知到对其感兴趣的订阅者后在 Broker 中的保留时间,在通知事件发生后 Broker 将在  $\Delta_i(x)$  时段内代替  $p_i$  重新发布  $x$ ,在这一时段内的  $tp_p$  时刻发生的发布事件称为持久发布事件  $e_B(\text{pub}(x), tp_p)$ ,而在事件  $e_B(\text{pub}(x), tp_p)$  发生后的  $x$  网络传播延迟时段内,  $tp_n$  时刻发生的通知事件称为持久通知事件  $e_j(\text{ntfy}(x), tp_n)$ ,其中  $j \in P, j \neq i$ 。

如图 4 所示一个基于 Broker 信息保持的发布/订阅系统。 $p_j$  在  $tp_1$  时刻发布信息  $x$ ,DNS 中 Broker $p$  在  $tn_1$  时刻向  $p_k$  发放投递信息  $x$ , $p_k$  上产生  $e_k(\text{ntfy}(x), tn_1)$  事件,但  $p_k$  未得到投递信息  $x$ 。为此,在  $tp_{p1} \in [tn_1, tn_1 + \Delta_i(x)]$  时刻,Broker $p$  代替  $p_j$  重新发布  $x$ ,发生持续发布事件  $e_B(\text{pub}(x), tp_{p1})$ ,并在  $tp_{n1} \in [tp_{p1}, tp_{p1} + D_{\text{diff}}]$  时刻发生持久通知事件  $e_k(\text{ntfy}(x), tp_{n1})$ , $p_k$  得到投递信息  $x$ ,但由于  $p_k$  处于非稳定期, $tp_{n1}$  时刻的投递信息  $x$  是不可靠的。 $p_j$  在  $tp_2$  时刻发布信息  $y$ ,Broker $p$  分别在  $tp_{p2} \in [tn_2, tn_2 + \Delta_j(y)]$  时刻,代替  $p_j$  重新发布  $y$ ,发生持久发布事件  $e_B(\text{pub}(y), tp_{p2})$ 、 $e_B(\text{pub}(y), tp_{p3})$ ,并在  $tp_{n2} \in [tp_{p2}, tp_{p2} + T_{\text{diff}}]$  时刻发生持久通知事件  $e_k(\text{ntfy}(y), tp_{n2})$ 。此时  $p_k$  处于稳定订阅时段内,投递信息  $y$  是可靠的。 $p_k$  如果在  $tn_1$  时刻没有得到投递信息  $x$ ,可以经 Broker $p$  代替  $p_j$  重新发布  $x$  后,在  $tp_{n1}$  时刻获得  $x$ ,但此时  $x$  是不可靠投递信息,可再经 Broker $p$  代替  $p_j$  重新发布  $y$  后,在  $tp_{n2}$  时刻获得可靠投递信息  $y$ 。

可见,如果 Broker 具有一定信息保持能力将提升信息投递的 QoS 保证。信息在 DNS 中的保留时间越长,DNS 中保留的信息越多,订阅者得到投递信息的可能性越大,也越容易保

证信息投递的 QoS;如果信息的保留时间足够长、DNS 中保留的信息足够多,所有订阅者都有可能得到投递信息,则可完全保证信息投递的 QoS,但必须以必要内存开销为代价。实际应用中可针对不同的 QoS 需求,选取不同的实现风格,并对信息的保留时间、数量与存储代价做折中处理。

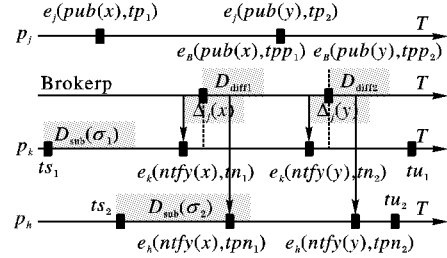


图 4 基于 Broker 信息保持的 Pub/Sub 系统

#### 4.2 Broker 结构的保证

合理地设计 Broker 结构也可保证信息投递的 QoS。图 5 是一种保证信息可靠投递的 Broker 结构设计,其主要组件及功能为:1)事件检测器检测订阅者、发布者上发生的事件;2)信息处理器处理异步消息,并将消息内容序列化;3)优先消息队列缓冲发布信息和订阅,相应分为发布队列 PMQ 和订阅队列 SMQ;4)查询引擎查询并反序列化消息,分解复合订阅,提取消息中的订阅与发布信息;5)匹配引擎主要负责在订阅表中注册订阅,根据匹配算法对发布与订阅进行匹配;6)具备垃圾回收机制的、基于堆结构的共享动态内存保存信息,回收废弃信息并根据路由表执行信息的转发;7)信息通知器向订阅者发放投递信息。

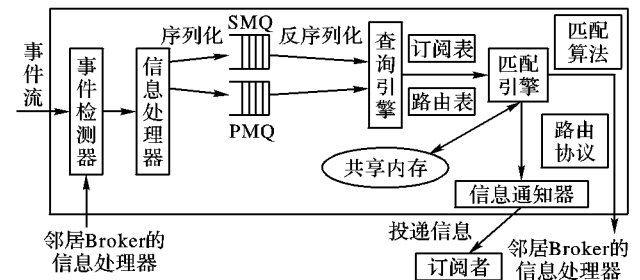


图 5 保证信息可靠投递的发布/订阅代理

图 5 中 Broker 的控制和数据流描述为:事件检测器从外部事件流中事件的发生,为事件设置时间戳,并触发一个事件消息,将消息发送到信息处理器;信息处理器序列化订阅  $\sigma$  和发布信息  $x$ ,并分别输入到优先消息队列 PMQ 和 SMQ;PMQ 和 SMQ 缓冲发布信息和订阅,可实现异步消息的序列化;查询引擎查询队列中的消息,经反序列化后提取消息中的订阅与发布信息,并输入至匹配引擎;匹配引擎根据匹配算法对  $x$  和  $\sigma$  进行匹配:如果匹配成功,则不保存  $\sigma$ ,并将  $x$  分别输入信息通知器和共享内存,由信息通知器向订阅者发放投递信息  $x$ ,共享内存将  $x$  保留一定时段,待重新发布;如果匹配失败,将  $\sigma$  注册到订阅表中,并将  $x$  转发到邻居 Broker 的信息处理器中。

图 5 中 Broker 对信息投递 QoS 的保证可通过图 6 说明。如图 6,进程  $p_k$  在订阅非稳定期得到投递信息  $x$ ,但这是一个无效投递,在  $tn_1$  不变的情况下,可以采取三种措施保证  $x$  的可靠投递:

1) 减小  $D_{\text{sub}}(\sigma_1)$ ,使得  $ts_1 + D_{\text{sub}}(\sigma_1) < tn_1$ ,由于订阅在优先队列中的滞留时间直接决定了订阅  $\sigma_1$  在订阅表中注册的时间延迟  $D_{\text{sub}}(\sigma_1)$ ,因此可将  $\sigma_1$  从图 5 中 SMQ 优先取出输入匹配引擎;

