

基于 DDS 的发布/订阅中间件设计

曹万华, 谢 蓓, 吴海昕, 程 雄

(武汉数字工程研究所, 武汉 430074)

摘要: 基于 OMG 组织颁布的实时系统中数据分发服务最新标准设计了一个新的发布/订阅中间件, 实现了分布式环境中以数据为中心的数据实时传输, 解决了分布式实时系统中服务质量约束和单点失效等问题。提出一种全局数据空间管理策略, 解决了通信节点动态增删问题。该模型与 CORBA 规范和其他发布/订阅模型相比, 具有较好的灵活性、实时性、可扩展性和可靠性。

关键词: 发布/订阅; 中间件; 数据分发服务; 服务质量

Design of Publish/Subscribe Middleware Based on DDS

CAO Wan-hua, XIE Bei, WU Hai-xin, CHENG Xiong

(Wuhan Digital Engineering Institute, Wuhan 430074)

【Abstract】 This paper designs a new publish/subscribe middleware based on data distribution service(DDS), which is published by OMG. Data centric-data real-time transferring in distributed environment is realized, and QoS restriction and a single point of failure are solved. Dynamical adding and deleting of communication nodes are realized by managing global data space in a special way. Compared with CORBA and other publish/subscribe models, this middleware is more flexible, realtime, scalable and reliable.

【Key words】 publish/subscribe; middleware; data distribution service(DDS); QoS

1 概述

消息中间件包括点到点、消息队列和发布/订阅 3 种工作模式。点到点模式具有很强的时间和空间耦合性, 使得通信灵活性受到很大限制。消息队列模式通过一个消息队列来传递消息, 解决了通信双方时间和空间松耦合的问题, 但不能实现消息消费者通信的异步^[1], 并且还存在着服务器瓶颈和单点失效的问题, 可靠性得不到保障。发布/订阅模型中发布者与订阅者通过主题相关联, 双方不必知道对方在何处, 也不必同时在线, 实现了通信双方时间、空间和数据通信的多维松耦合。表 1 是这 3 种通信模式松耦合的比较。

表 1 几种通信模式的松耦合性

通信模式	时间松耦合	空间松耦合	数据传输松耦合
点到点	否	否	否
消息队列	是	是	仅生产者支持
发布/订阅	是	是	是

已有的发布/订阅中间件产品很多, 如 TIB/Rendezvous, Gryphon 等, 但是始终缺乏一个统一的标准, 各企业之间的接口各异, 采用的通信协议不统一, 导致了应用的局限性。SUN 公司发布的 Java 消息服务(Java message service, JMS)规范基本解决了各厂商产品之间的互联问题, 但是该规范用在分布式实时环境中还存在很多问题: 它采用 Java 实现, 影响了系统的实时性; JMS 中并没有提出 QoS 约束, 用户不能根据自身需要指定通信方式, 限制了通信的灵活性; 在 JMS 中采用固定的节点服务器保存主题或队列, 不能均衡负载或失败转移节点服务器, 存在服务器瓶颈和安全隐患。

数据分发服务(data distribution service, DDS)^[2]是OMG继推出CORBA^[3]规范后, 专门针对分布式实时系统中数据发布

/订阅模型颁布的一个最新规范, 它纯粹以数据为中心来进行数据分发, 并将资源状况、对资源的期待程度、网络状况等都用QoS参数来描述, 大大增强了通信的实时性和灵活性。

2 基于 DDS 的发布/订阅模型

本文基于 DDS 规范, 提出了一个新的发布/订阅模型。 $PS(P,S)$ 表示一个发布-订阅模型, 其中, P 为发布者集合; S 为订阅者集合。

发布者集合由若干发布者构成, $P = \{P_1, P_2, \dots, P_n\}$ 。发布者负责发布数据, 通过创建不同的数据写入者来发布不同类型的数据, 每个数据类型用一个主题来标识。发布者并不实际传送数据, 只创建和管理数据写入者, 由数据写入者实际传送数据给订阅方。

$$P_i = \{DW_{i1}, DW_{i2}, \dots, DW_{in}\}$$

其中, DW_{ij} 表示一个数据写入者, 它负责发布主题, 接收该主题的订阅消息, 并传送该主题下的数据给订阅方。数据写入者由创建它的发布者来管理, 每个数据写入者发送指定数据类型的数据。

$$T = \{T_1, T_2, \dots, T_n\}$$

其中, T 为主题集; T_i 为一个主题; $Q = (x, y)$ 为 x 对 y 的QoS约束, 具体包括时延(Deadline)、持久性(Durability)、可靠性(Reliability)和优先级(Priority)。

一个数据写入者与一个主题对应, $f: DW_{ij} \rightarrow T_k$, 当然,

基金项目:“十一五”国防预研基金资助项目

作者简介:曹万华(1966-), 男, 研究员, 主研方向: 软件工程技术及应用, 构件与中间件技术; 谢 蓓, 硕士研究生; 吴海昕, 工程师; 程 雄, 硕士、助理工程师

收稿日期: 2006-09-25 E-mail: nory889@163.com

可能存在多个数据写入者发布同一主题。

订阅者集合由若干订阅者构成， $S = \{S_1, S_2, \dots, S_n\}$ 。订阅者负责订阅并接收数据，通过创建不同的数据读入者来订阅不同的数据类型，每个数据类型同样用一个主题来标识。

$$S_i = \{DR_{i1}, DR_{i2}, \dots, DR_{in}\}$$

其中， DR_{ij} 表示一个数据读入者。

一个数据读入者与一个主题对应， $g: DR_{ij} \rightarrow T_k$ 。多个数据读入者可以同时订阅同一主题。

当 $f(DW_i) = g(DR_j) = T$ ，且 $Q(DW_i, T) = Q(DR_j, T)$ 时，即数据写入者发布的主题与数据读入者订阅的主题相同，且 QoS 匹配时就形成发布-订阅关系，记为 $R_PS(DW_i, DR_j, T)$ 。数据读入者只能订阅数据写入者已发布的主题， $g(S) \subseteq f(P)$ 。

对于一个待订阅的主题，同一时刻可能有多个满足条件的数据写入者存在，数据读入者可以按照一定策略选择一个数据写入者进行订阅，当该数据写入者失效时，可用其他的数据写入者替换，这样可有效避免单点失效的问题。

可见，DDS发布/订阅模型中发布者和订阅者是通过主题相关联的。当订阅和发布的主题相同及相应QoS匹配时，全局数据空间(global data space, GDS)^[2]通知发布者和订阅者做好准备，进行数据传输。图1是该模型的逻辑示意图。

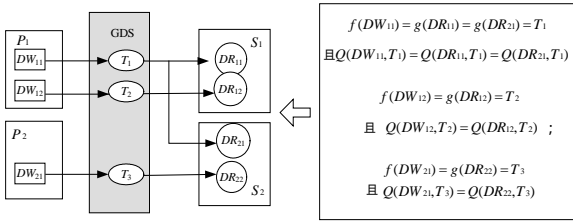


图1 DDS发布/订阅模型逻辑示意图

3 全局数据空间的组织与管理

3.1 全局数据空间的结构

DDS发布/订阅中间件中主题的管理与匹配、发布者和订阅者连接事件的通知都由GDS来完成，它是本模型最核心的部分。GDS由一张全局一致的发布主题表、一张订阅登记表、发布数据缓冲区和订阅失败队列4部分构成。

(1)发布主题表

$PT_Table(ID, P_IP, Durability, Reliability, Deadline)$ 反映了所有发布主题的状态信息，其中， ID 为发布的主题； P_IP 为发布者地址； $Durability$ 为数据持久性，0表示非持久性数据，无须保存，1表示持久性数据，需要保存； $Reliability$ 为传输方式的可靠性， $Best-effort$ 表示采用尽力方式传输， $Reliable$ 表示采用可靠方式传输； $Deadline$ 定义了发布的最大时延。所有节点上的发布主题表保持一致并实时更新。

(2)订阅登记表

$SR_Table(ID, S_IP, Reliability, Deadline, Priority, PID)$ 反映了当前订阅该节点的订阅信息，其中， ID 为订阅主题； S_IP 为订阅者地址； $Reliability$ 为订阅方要求的传输可靠性； $Deadline$ 为订阅者的传输最大时延； $Priority$ 表示订阅事件的优先级，值越小其优先级越高； PID 为订阅线程的PID。当建立发布-订阅关系时，订阅者的 $Deadline$ 小于发布者的 $Deadline$ 。不同节点的订阅登记表各不相同。

节点在订阅主题时，通过发布主题表来确定提供该订阅主题；在发布主题时，通过订阅登记表来确定数据该发往何处。

(3)发布数据缓冲区

$Data_pool$ 保存了由本节点发布且需要保存的发送数据。其结构如图2所示，数组的每一项表示同一主题已发送的数据。GDS根据每种主题下数据的 $Livespan$ 来决定何时更新缓冲区。

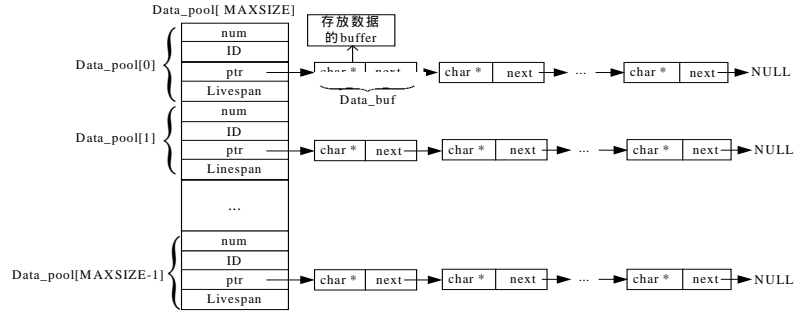


图2 发布数据缓冲区结构示意图

(4)订阅失败队列

Sub_fail_que 保存本节点向其他节点订阅失败的历史记录，每个节点表示在同一主机上订阅失败的记录。其结构与图2类似。

3.2 全局数据空间的管理策略

GDS是本模型实现主题订阅和数据分发的关键。如何更新各个节点上的发布主题表使其保持动态一致，以及维护和更新订阅登记表是GDS管理的核心环节。本文提出了5条GDS管理规则(见图3)。

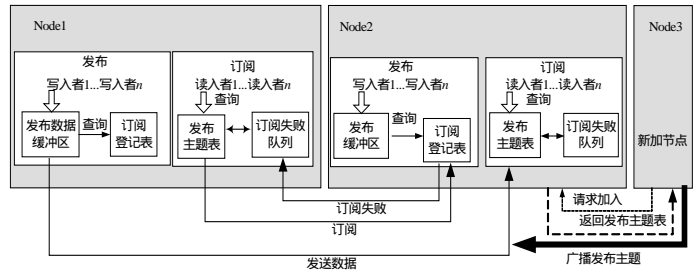


图3 GDS更新策略

规则1 每个节点开机时向所有节点广播开机信息，收到开机信息后，各个节点检查自己的订阅失败队列，将在开机节点上订阅失败的订阅消息重新发给开机节点，订阅成功后从队列中删除。

规则2 订阅方根据发布主题表向发布者发送订阅消息，若订阅成功就在发布者的订阅登记表中登记订阅记录。发布节点上的GDS在发布数据缓冲区查找是否存在该订阅者所订主题的数据，然后激活相应数据写入者将数据发送给订阅方的相应数据读入者。若订阅失败就将该订阅记录记入本节点的订阅失败队列。

规则3 订阅者变更订阅记录时通知相应的发布者更新其订阅登记表。发布者变更发布主题时需要通知所有节点更新发布主题表，以保持发布主题表的全局一致性。

规则4 当有新节点加入时，新节点向其邻近节点发送“加入请求”，邻近节点若接受请求就将本节点上的“发布主题表”发送给新加入节点。

规则 5 节点离开或失效的情况：当节点离开时，向其他节点发送离线消息，其他节点删除订阅登记表中离线节点的订阅记录，离线节点同时清空本节点上的订阅登记表；各个节点通过定期向其他节点发送心跳信息来表示该节点还“活着”。一段时间内收不到某节点的心跳信息表明该节点已“死亡”，其他节点要删除订阅登记表中“死亡”节点的订阅记录。

4 发布/订阅过程

DDS 发布订阅中间件主要由发布者、订阅者、数据写入者、数据读入者和 GDS 来实现主题的发布订阅和数据传输功能。以下将从发布过程、订阅过程、数据分发过程等方面进行阐述。

4.1 发布过程

图 4 给出了 DDS 发布/订阅消息中间件的工作流程。当节点 Nodi 要发布消息时，由发布者 P_i 创建一个数据写入者 DW_i ， DW_i 将要发布消息的主题 T_i 及其可以提供的服务质量 $QoS = Q(DW_i, T_i)$ 发布到其他各节点的 GDS 上，发布成功后 DW_i 进入阻塞状态。各个节点在其 PT_Table 中增加 DW_i 的发布记录 $(T_i, P_i_IP, Durability, Reliability, Deadline)$ 。

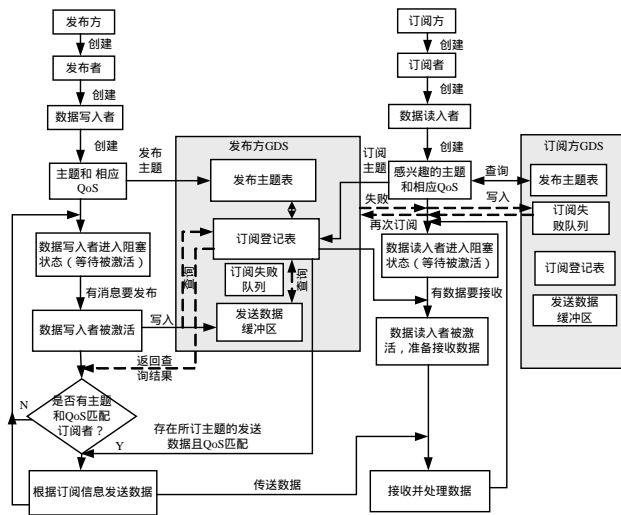


图 4 DDS 发布/订阅中间件的工作流程

4.2 订阅过程

节点 Nodj 要订阅主题为 T_j 的数据时，由订阅者 S_j 创建一个数据读入者 DR_j ， DR_j 首先在本节点的 PT_Table 中找到相匹配的发布者，按照一定策略从中选择一个作为本次订阅的对象，然后将 T_j 及 QoS 发送到该发布节点的 GDS 上，接着进入阻塞状态等待有数据时被激活。

节点 Nodi 收到 DR_j 的订阅信息时，先在 GDS 的 SR_Table 添加订阅记录 $(T_i, S_i_IP, Reliability, Deadline, Priority, PID_j)$ ，然后查看 PT_Table 中 T_j 的 Durability 值。

若 $Durability=0$ ，表示该主题的数据是非持久性数据，没有保存在 Data_pool 中。若 $Durability=1$ ，表示发布数据缓冲区有可能存在 T_j 对应的数据，GDS 在 Data_pool 中查找满足条件 $ID=T_j$ ，且相应的 $num > 0$ 的项，若存在则进行 QoS 匹配，激活数据写入者 DW_i ， DW_i 将该项链表中保存的所有数据按照相应的 QoS 发送给订阅方 PID_j 。

4.3 数据分发过程

(1) 发布者发布数据

在 Nodi 上，当发布的主题 T_i 有数据要发送时，数据写入者 DW_i 被激活，它先查看 PT_Table 中 T_i 所对应的 Durability 值。

若 $Durability=1$ ，则在 Data_pool 中 $ID=T_i$ 的链表中添加一项，以便后来的订阅者可以收到该数据；否则根据下面的步骤发送完数据后将其丢弃。

(2) 传输数据

Nodi 中的 GDS 在 SR_Table 查找是否存在满足条件 $f(DW_i) = g(DR_j)$ ，且 $Q(DW_i, f(DW_i)) = Q(DR_j, g(DR_j))$ 的数据读入者 DR_j 。

若存在 $DR_{j_1}, DR_{j_2}, \dots, DR_{j_n}$ ，假定 DW_i 发布的主题为 T ，则 DW_i 就和 $DR_{j_1}, DR_{j_2}, \dots, DR_{j_n}$ 形成了发布订阅关系 $R_PS(DW_i, DR_{j_1}, T), \dots, R_PS(DW_i, DR_{j_n}, T)$ 。

GDS 将 SR_Table 中 $DR_{j_1}, DR_{j_2}, \dots, DR_{j_n}$ 的订阅信息发送给数据写入者 DW_i ，并激活 $DR_{j_1}, DR_{j_2}, \dots, DR_{j_n}$ 对应的 $DR_{j_1}, DR_{j_2}, \dots, DR_{j_n}$ 。 DW_i 根据订阅信息按照相应的 QoS 策略发送数据给 $DR_{j_1}, DR_{j_2}, \dots, DR_{j_n}$ ，然后进入阻塞状态等待下一次被激活。若不存在满足上述条件的数据读入者，则进入阻塞状态等待被激活。数据读入者接收到数据后对数据进行实际处理，然后进入阻塞状态，等待下一次被激活。

该模型将发布方可提供资源状况及订阅方对资源的期待程度分别用可靠性、持久性、最大时延和生命周期等 QoS 参数来描述。中间件通过这些参数选择最符合通信双方 QoS 要求的传输方式来分发数据，既实现了数据传输的实时性，也增加了通信的灵活性。总之，该中间件设计方案遵循消息中间件的思想，以消息流作为整个平台的触发点和导向，实现数据的实时分发。

5 性能分析与比较

5.1 通信特点

该 DDS 发布/订阅中间件采取分布式体系结构，发布者和订阅者通过指定通信的 QoS 来选择合适的通信方式，实现了发布方和订阅方的异步实时通信。具体特点如下：

- (1) 便捷性。订阅者根据发布主题表可以随时订阅自己需要的主题，而不用管发布者是谁。一旦订阅成功，就形成了发布-订阅关系，实现了一个数据“推模型”^[3]，订阅者就只管接收数据。
- (2) 实时性。发布者和订阅者通过 QoS 约束，可以选择最符合自己要求的传输方式。
- (3) 动态性。根据系统需要可以随时添加或删除节点。
- (4) 可靠性。真正实现了全分布式结构，单点失效对系统功能无影响。
- (5) 可扩展性。可以随时增加一个新主题，为系统添加新功能。
- (6) 灵活性。根据 QoS 可以采用 UDP、RUDP、TCP 和组播等多种传输策略。

5.2 DDS 规范与 CORBA 规范的比较

DDS 和 CORBA 都是 OMG 组织颁布的分布式系统数据传输规范，DDS 是 CORBA 的一个对等体，是在对实时 CORBA、NDDS 应用的基础上发展而来的^[4]。二者之间既存在联系又有很大的差别，其比较如表 2 所示。

表 2 DDS 与 CORBA 的比较

比较项目	DDS	CORBA
通信模式	以数据为中心的发布-订阅模式	以对象为中心的 C/S 模式
数据传输机制	每个发布-订阅就是一个数据流，直接传送订阅者感兴趣的数据	通过对象请求代理调用远程对象方法得到自己感兴趣的数据
适用范围	对 QoS 有所要求，数据传输频繁且参与者众多的高效实时数据分发的分布式系统 ^[4]	应答-请求服务、文件系统和事件处理等传输数据量大且实时性要求不高的分布式系统

(下转第 83 页)