

# 基于 DDS 的实时数据分发系统的研究与实现

刘芳明

北京邮电大学 电信工程学院, 北京 (100876)

E-mail: fantacylfm@gmail.com

**摘要:** 在重点分析对象管理组织(OMG)颁布的数据分布服务(Data Distribution Service, DDS)采用的以数据为中心的发布/订阅(DCPS)模型基础上,设计实现了一个采用发布/订阅机制,基于主题和QoS控制策略的实时数据分发系统。

**关键词:** 数据分发服务; 发布/订阅; 实时数据分发; CORBA

**中图分类号:** TP319

## 1. 引言

随着Internet技术的广泛应用和计算机技术的飞速发展,各种应用系统的体系结构呈现出以网络为中心的趋势,这便对通信的实时性、动态灵活性提出了更高的要求,同时要求分布式系统各参与者之间采用一种具有松散耦合特性和QoS支持的灵活通信模型和交互机制。而现有的发布/订阅系统虽然提供了良好的开发平台和通信支持,却缺乏合适的系统控制接口和应用级QoS的有效支持,并不能很好地解决通信的实时性、快捷性和灵活性问题。如当前较为流行的实时CORBA<sup>[1]</sup>技术,由于它是以对象和服务为中心,采用了C/S通信模式,通信机制较为复杂,数据收发需要建立连接的过程,不能完全满足系统对实时性能的需要,并且它没有QoS支持,不能满足通信灵活性要求。JAVA消息服务JMS<sup>[2]</sup>包含点对点和发布/订阅两种消息模型,提供可靠消息传输、事务和消息过滤等机制,适合大规模以数据为中心的网络,但是它缺乏应用级QoS策略,仍然不适合实时性要求高和紧急任务系统,例如:医疗系统、舰船控制系统和遥感勘测系统等领域。因此急需一种能为实时系统应用开发者提供高级抽象接口的同时还能有效合理地控制部署实时系统所需的QoS参数的系统来满足分布式实时应用需求。

OMG在实时CORBA的基础上,根据以数据为中心的发布/订阅(DCPS)模型制定了DDS<sup>[3]</sup>数据分发服务规范。DDS信息分发中间件是一种轻便的、能够提供实时信息传送的中间件技术。目前,信息分发中间件产品已经在发达国家中的舰船控制、数字电视系统、飞行模拟器和军队网络系统中广泛使用。网络信息分发服务NDDS<sup>[4]</sup>是RTI公司的信息分发中间件产品,完全符合OMG组织的DDS标准。RTI公司的NDDS中间件已经和DDS API相融合,可以提供差错避免,减少风险和系统开发费用。但是NDDS只是有限考虑了应用级QoS的设置,但并未有效使用应用级的QoS控制发布/订阅信息。

在保持性能和扩展性的前提下,本文基于 DDS 规范实现了一个以数据为中心的实时数据分发系统,实现了基于主题的数据发布/订阅功能,并能通过 QoS 分级和分类来控制通信质量。

## 2. DDS 数据分发服务概述

DDS 规范标准化了分布式实时系统中数据发布、传递和接收的接口和行为,定义了以数据为中心的发布/订阅(Data-Centric Publish Subscribe)机制,提供了一个与平台无关的数据模型,该模型能够映射到各种具体的平台和编程语言。DDS 规范的目的是简化分布式系统中数据的有效发布,它主要应用在高性能、可预见性和对资源有效使用的关键任务领域。

DDS 规范描述了两个层次的接口：

低层 DCPS（Data-Centric Publish-Subscribe）：将数据发布者发布的信息高效准确地传送给数据订阅者。

可选的高层 DLRL（Data Local Reconstruction Layer）：允许将服务简单的集成到应用层。

DCPS 层是 DDS 规范的核心，它提供了数据发布的基础架构。DLRL 层建立在 DCPS 基础之上，主要是规定了应用层和 DCPS 层之间的接口，该接口将接收到的数据进行融合并传送给应用层，通过底层的 DCPS 提供服务，简化了编程工作。本文实现了 DCPS 层的数据分发服务。

### 3. DCPS 层基本机制

DCPS 是建立在“全局数据空间”（Global Data Space）概念基础上的，这个全局数据空间是可以为感兴趣的应用程序所访问的。DCPS 模型又分为平台独立模型 PIM 和平台专门模型 PSM，PSM 是以 PIM 为基础的 OMG IDL 平台。PIM 平台总体框架结构如图 1 所示。

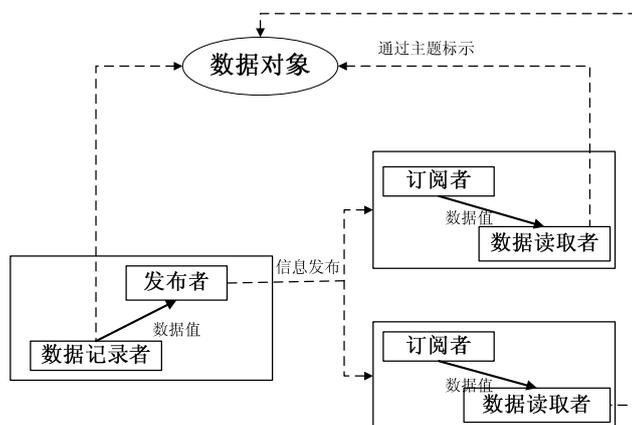


图 1: PIM 平台框架结构

PIM 平台主要由域、发布者、订阅者、数据写入者、数据读取者和数据对象几个实体对象组成。域（Domain）是 DCPS 中的基本分割单位。任意其它实体都属于某个域并且只能与位于相同域的其他实体相互作用。应用程序通过属于相应域的实体可以作用于某个域。一个域中的每个过程叫做域参与者。提供数据的应用者通过一定的规则发布信息，中间件（middleware）中发布信息的单元称之为发布者；需要接收指定数据的单元称为订阅者。当发布者将数据发布到全局数据空间后，中间件负责将这些信息传播给所有感兴趣的订阅者。

使用 DDS 服务的用户能够使用数据记录者和数据读取者的类型化接口有效的“读、写”数据。每个数据记录者/数据读取者与一特定主题绑定。数据记录者负责整编（marshaling）数据以及向发布端传输数据。数据读取者（DataReader）从订阅端获取数据，按特定主题分解数据，整理成恰当类型，并将实例传递给应用程序。在底层，DDS 中间件将会负责实时地分发数据，而其它的对某些数据感兴趣的能够实时地存取“最新的”数据。

DCPS 层提供了应用程序所需的发布和订阅数据的功能。DDS 提供了基于主题的发布/订阅模式，发布方在中间件上发布一个主题后，订阅方可以根据主题来查阅、获取发布方发布的信息。同时 DDS 提供了对基于内容的发布/订阅的支持。通过设置通信信息的关键字，可以进一步过滤数据，进而减少网络的通信量。本文采用基于主题的发布/订阅模式。

发布方和订阅方之间的连接是否建立是由中间件服务的检查机制来决定的，服务不仅检

查发布方发布的主题是否满足订阅方的要求（包括主题名称和数据类型），并且检查其 QoS 策略是否兼容。QoS 策略就是可以控制数据分发服务的一系列特性的组合。系统中的所有实体都具有 QoS 策略，包括主题、数据记录者、数据读取者、发布者、订阅者和域参与者。

为了保证通信高效可靠的进行，数据发布方的 QoS 策略必须和数据接收方的 QoS 策略相符。若订阅者请求可靠的信息数据，而相应的发布者却仅仅提供了尽力而为的服务质量策略，则实体之间不会建立通信。为了解决此问题，需要尽量保证发布者和订阅者的去耦合，订阅方对相关服务质量策略指定一个申请值，而发布方则对该服务质量策略指定一个供应值。系统确定发布方的供应值和订阅方的申请值是否一致，如果双方的服务质量策略是一致的，则通信正常建立；如果不一致，则系统服务不会在这两个实体对象之间建立通信，并会通知发送方和接收方。

发布端/订阅端获取信息有两种通信模式：异步方式和同步方式。在异步模式下，DDS 提供监听（Listener）接口，当相应的信息出现时，中间件通知 Listener 接口，Listener 通过相应的回调（callback）来通知应用程序作相应处理。在同步模式下，应用程序调用 WaitSet 接口，并停止等待，直到最新的数据信息到达。本文采用异步模式。

## 4. 系统的基本结构及实现

### 4.1 系统基本结构及实现

本文实现了 DCPS 层的 PIM 平台，根据规范将信息分发中间件系统分为五个模块，分别是基础模块、主题模块、发布模块、订阅模块和域模块，如图 2 所示。

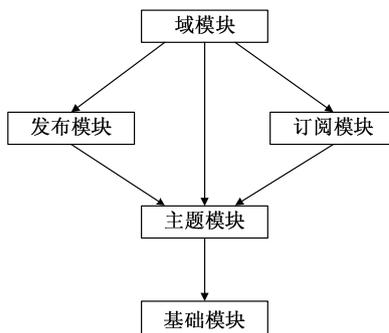


图 2: 系统模块组成

- 基础模块：定义了所有模块的抽象类和接口，还为中间件的两个交互类型（包括通知类型和等待类型）提供支持。
- 域模块：域模块包含多个域参与者类，该类可作为中间件服务的入口点，也可以作为其它类的工厂模式。
- 主题模块：包括主题类、主题内容过滤类、复合主题类、主题监听接口以及主题的 QoS 策略。
- 发布模块：包括发布者类、数据写入者类、发布监听接口、数据记录监听接口。
- 订阅模块：包括订阅者类、数据读取者类、读取条件类和查询条件类以及订阅监听接口和数据读取监听接口。

### 4.2 数据发布-订阅流程及实现

发布订阅时序如图 3 所示：

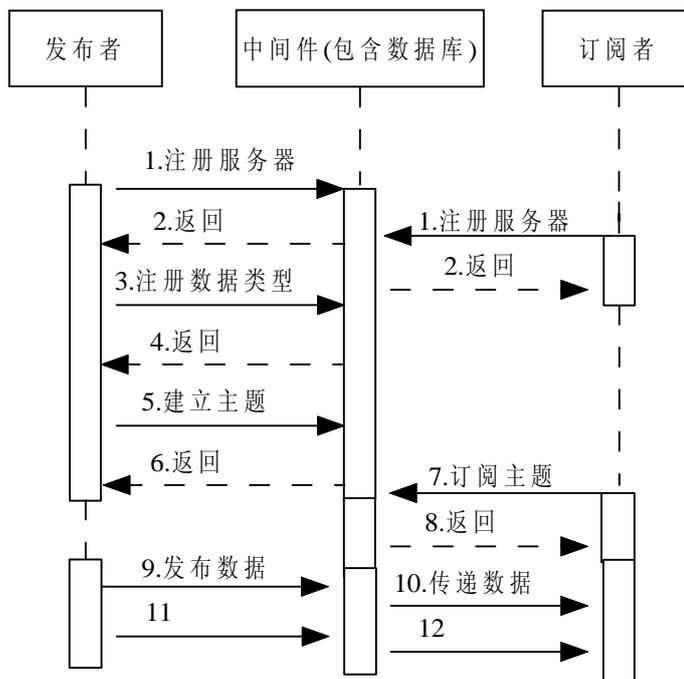


图3 发布/订阅时序图

(1) 首先将 DCPS 信息仓库建立起来，并作相应的初始化工作，包括域的建立，QoS 和传输的初始化。然后发布者/订阅者端建立域工厂 (DomainParticipantFactory)，由域工厂建立域参与者，各个域之间是靠 id 来区分的，发布者/订阅者的域 id 必须与 DCPS 层建立的域 id 一致才能进行通信。

(2) 发布者在中间件注册数据类型 DataType，然后设置主题 QoS 便能建立主题。同样的订阅者设置 QoS 也能建立主题。主题建立后，经过中间件服务的检查机制，若符合连接要求发布者/订阅者便建立连接。

(3) 连接建立后发布者设置数据记录者 QoS 并建立数据记录者发布最新的数据，订阅者设置数据读取者 QoS 并建立数据读取者接收最新的数据。

#### 4.2.1 发布的实现

1. 首先初始化域参与者:

```

DDS::DomainParticipantFactory_var dpf = TheParticipantFactoryWithArgs(argc, argv);
DDS::DomainParticipant_var participant = dpf->create_participant(411, //域 ID
PARTICIPANT_QOS_DEFAULT, //参与者 QoS
DDS::DomainParticipantListener::_nil());
    
```

2. 注册数据类型:

```

MessageTypeSupportImpl* servant = new MessageTypeSupportImpl();
if (DDS::RETCODE_OK != servant->register_type(participant.in (), "Message")) {
std::cerr << "register_type failed." << std::endl;
return 1;
}
    
```

3. 建立主题:

```

DDS::TopicQos topic_qos;
participant->get_default_topic_qos(topic_qos);
. . . //QoS 设置部分
    
```

```

DDS::Topic_var topic =
participant->create_topic ("Movie Discussion List", //主题名称
"Message", //数据类型
topic_qos, //主题 QoS
DDS::TopicListener::_nil());

```

4. 初始化传输：传输设置采用文件配置机制，包括 TCP,UDP 选择等

```

const TAO::DCPS::TransportIdType TRANSPORT_IMPL_ID = 1;
TAO::DCPS::TransportImpl_rch tcp_impl =
    TheTransportFactory->create_transport_impl (TRANSPORT_IMPL_ID,
    TAO::DCPS::AUTO_CONFIG);

```

5. 建立发布者，并将发布者连接至传输层上：

```

DDS::Publisher_var pub =participant->create_publisher(PUBLISHER_QOS_DEFAULT,
    DDS::PublisherListener::_nil());
TAO::DCPS::PublisherImpl* pub_impl =
    reference_to_servant< TAO::DCPS::PublisherImpl,

```

```

DDS::Publisher_ptr>(pub);

```

```

TAO::DCPS::AttachStatus status =pub_impl->attach_transport(transport_impl.in());

```

6. 建立数据记录者：

```

DDS::DataWriterQos dw_qos;
pub->get_default_datawriter_qos (dw_qos);
DDS::DataWriter_var dw =pub->create_datawriter(topic.in (), //与之对应的主题
    dw_qos, //数据记录者 QoS
    DDS::DataWriterListener::_nil());

```

7. 写数据：

```

Writer *writer = new Writer(dw.in());
writer->start ();
这里 writer->start ()既是执行了以下操作：
MessageDataWriter_var message_dw = MessageDataWriter::_narrow(writer.in());
Messenger::Message message;
DDS::InstanceHandle_t handle = message_dw->_cxx_register (message);
DDS::ReturnCode_t ret = message_dw->write(message, handle);

```

#### 4.2.2 订阅的实现

1. 订阅端初始化域参与者，注册数据类型，建立主题，初始化传输与发布端一致，然后需要建立发布者，并将发布者连接至传输层上，其操作如下：

```

DDS::Subscriber_var sub =participant->create_subscriber(SUBSCRIBER_QOS_DEFAULT,
    DDS::SubscriberListener::_nil());
TAO::DCPS::SubscriberImpl* sub_impl =reference_to_servant<
TAO::DCPS::SubscriberImpl,
    DDS::Subscriber_ptr> (sub.in());

```

```

TAO::DCPS::AttachStatus status =sub_impl->attach_transport(transport_impl.in());

```

2. 激活 Listener：

```

DataReaderListenerImpl listener_servant;
PortableServer::POA_var poa = TheServiceParticipant->the_poa ();
CORBA::Object_var obj = poa->servant_to_reference(&listener_servant);

```

```
DDS::DataReaderListener_var listener =DDS::DataReaderListener::_narrow (obj.in ());
```

3. 建立数据读取者:

```
DDS::DataReaderQos dr_qos;
```

```
sub->get_default_datareader_qos (dr_qos);
```

```
DDS::DataReader_var dr = sub->create_datareader(topic.in (),dr_qos,listener.in ());
```

//topic.in (): 与数据读取者对应的主题; dr\_qos: 数据读取者 QoS

4. DataReaderListener 接口定义:

```
interface DataReaderListener : Listener {  
    void on_requested_deadline_missed(in DataReader reader,  
        in RequestedDeadlineMissedStatus status);  
    void on_requested_incompatible_qos(in DataReader reader,  
        in RequestedIncompatibleQosStatus status);  
    void on_sample_rejected(in DataReader reader, in SampleRejectedStatus status);  
    void on_liveliness_changed(in DataReader reader,  
        in LivelinessChangedStatus status);  
    void on_data_available(in DataReader reader);  
    void on_subscription_match(in DataReader reader,  
        in SubscriptionMatchStatus status);  
    void on_sample_lost(in DataReader reader, in SampleLostStatus status);  
};
```

接收到数据会自动回调 on\_data\_available 方法对数据进行相应处理。

## 5. 总结

基于 DDS 设计的实时数据分发系统, 采用发布/订阅通信机制, 根据主题进行点到点消息发送, 有效地减少了不必要的网络流量, 通过提供应用级 QoS, 用户对发布/订阅的行为进行控制, 并且具有应用规定语义下的可靠性。今后的工作是深入研究 DDS 规范, 实现基于内容的发布订阅, 实现同步获取信息模式 (WaitSet), 加入对 XML 数据的支持, 完善 QoS 设置和过滤和调度处理, 进一步提高系统的整体性能。

## 参考文献

- [1]Object Management Group. Real-time CORBA Specification[R]. Version1.2, Jan.2005
- [2]Sun. JAVA Message Service[R]. Version1.1, April.2002
- [3]Object Management Group. Data Distribution Service for Real-time Systems Specification[R]. Version1.1, Nov.2005
- [4]Network Data Distribution Service (NDDS) Real - Time Publish - Subscribe Network Middleware [DB/ OL ]. <http://www.rti.com> 2007
- [4]Gerardo P C. OMG Data-Distribution Service (DDS): Architectural Update[C]. 2004 IEEE Military Communications Conference, 2004, 2: 961~967.
- [6]Burlingame, Gerardo P C, Hunt G A. DDS Enabling Global Data[R]. Real-Time Innovations, Inc., December 2005.
- [7]Giddings V. Tutorial on the OMG Data Distribution Service[R]. Objective Interface Systems, Inc., 2005.
- [8]Schneider S, Farabaugh B. Using the DDS Standard for High Reliability Applications[R]. Real-Time Innovations, Inc., 2004.

## Research on the Real-time Data Distribution System Based on DDS and Its Implementation

Liu Fangming

Dept. of Telecommunication Engineering, Beijing University of Posts and Telecommunications,  
Beijing (100876)

### Abstract

In this paper, the data-centric publish-subscribe model which is used by Data Distribution Service for Real-Time Systems Specification published by Object Management Group is analyzed. Then designs and implements a topic-based, QoS enabled real-time data distribution system which adopts the data-centric publish-subscribe model.

**Keywords:** *Data Distribution Service; publish/subscribe; real-time data distribution; CORBA*