数据分发服务
——以数据为中心的发布/订阅式通信

任昊利 等 编著
赵洪利 主审

清华大学出版社

数据分发服务
——以数据为中心的发布/订阅式通信

任昊利 李旺龙 张少扬 王恒佳 编著

清华大学出版社
北京

内 容 简 介

DDS(Data Distribution Service,数据分发服务)是OMG发布的有关分布式实时系统中数据传输的一个规范(2004年12月发布1.0版,2007年1月发布1.2版)。随着实时分布式系统复杂度的不断增加和研发规模的迅速扩大,系统集成的难度和风险都在大幅度提高,DDS为各种不同的分布式应用提供了数据通信模型,对分布式的异构系统集成提供了很好的解决方案。DDS是以数据为中心的发布/订阅通信模型,针对强实时系统进行了优化,提供低延迟、高吞吐量以及对实时性能的控制级别,从而使DDS能够广泛地用于航空航天、国防、分布式仿真、工业自动化、分布控制、机器人、电信以及物联网等多个领域。

本书的编写本着由浅入深、深入浅出的原则,系统地介绍了DDS的使用、编程技术。本书中配有大量的例子,供使用人员参考,适合软件工程师、软件架构师、软件项目经理等专业人员使用,同时也适合作为硕士研究生、本科生的教材。

图书在版编目(CIP)数据

数据分发服务——以数据为中心的发布/订阅式通信/任昊利等编著. —北京:清华大学出版社,2014
ISBN 978-7-302-38107-5

Ⅰ. ①数… Ⅱ. ①任… Ⅲ. ①数字信号处理 Ⅳ. ①TN911.72

中国版本图书馆CIP数据核字(2014)第224490号

责任编辑:袁勤勇　王冰飞
封面设计:傅瑞学
责任校对:梁　毅
责任印制:沈　露

出版发行:清华大学出版社
网　　址:http://www.tup.com.cn, http://www.wqbook.com
地　　址:北京清华大学学研大厦A座　　**邮　　编**:100084
社 总 机:010-62770175　　**邮　　购**:010-62786544
投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn
质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn
课件下载:http://www.tup.com.cn,010-62795954
印 刷 者:北京富博印刷有限公司
装 订 者:北京市密云县京文制本装订厂
经　　销:全国新华书店
开　　本:185mm×260mm　　**印　　张**:18　　**字　　数**:448千字
版　　次:2014年12月第1版　　**印　　次**:2014年12月第1次印刷
印　　数:1～2000
定　　价:39.00元

产品编号:057638-01

前言

编者从接触数据分发服务技术(DDS)到使用它做一些课题已经有几年了,在这期间逐渐感受到DDS的先进性并坚信它将广泛地应用于信息系统集成领域。在课题研究和学术交流的过程中经常会有人问“DDS到底是做什么的和有什么用?”每当遇到这样的问题,我就会感到有必要把DDS介绍给大家。

DDS能够满足很多领域的应用需要,从一开始就受到了业界的关注和多方支持,国际上已经展开了大量研究,美国国防部于2006年把DDS列为全球信息栅格(GIG)的数据分发标准,目前已在美国的军方得到了广泛应用。其实,在综合电子信息系统建设中遇到的很多问题都可以在DDS中找到借鉴的思路。

我国对DDS的研究和应用刚刚兴起,有很多工程师希望得到中文版本的技术参考书,然而目前市面上还没有介绍DDS的书籍,更谈不上“自主可控”的DDS产品,这更激发了编者编写本书的热情。

目前的DDS产品提供给用户的是函数库,对于很多工程人员来讲,想尽快上手并不是一件易事。受到DDS软总线思想的启发,我们开发了一个使用DDS构建分布式信息系统的工具,系统工程师借助该工具可以快速搭建起支持分布式异构系统的通信框架,可节省大量的系统开发、维护时间和费用;各专业的程序员也不需要对数据传输层做大量的基础性工作(例如编写、调试通信模块),而只需要把精力集中在本专业的业务上。对于该工具本书未作介绍,有兴趣的读者可与我们进行交流。

在本书编写的过程中,赵洪利部长给予了指导和建议,在此表示感谢。另外,感谢创景公司的杨亦忠先生为本书的出版提供的帮助。最后还要感谢我的家人和朋友,他们在精神和生活方面给予我大力支持,是我奋斗的动力和源泉。

尽管我们力争做到精益求精,但由于水平所限,错误在所难免,欢迎大家批评与指正。

任昊利

2014-5-16

于北京怀柔雁栖湖畔

目录

第1章 概　述

对分布式实时应用程序而言，以数据为中心的需求要比以服务为中心的需求多，这意味着在分布式系统中参与者的主要目标是传播应用程序数据而不是共享服务。应用程序的供应者和消费者的数据类型可以在设计时就知道，也可能在应用程序执行期间被改变。以数据为中心的共享模式有效地实现了发布/订阅通信模型，这种模式有别于请求/响应通信模型。

OMG 实时系统的数据分发服务（Data Distribution Service（DDS）for Real-Time Systems）与以数据为中心的分布式应用程序的性能要求和硬实时要求相适应。DDS 是用于分发实时应用程序数据的网络中间件，它简化了应用程序的开发、部署和维护，并在大量传输网络中提供了针对时间关键数据的快速且可预见的分发。

DDS 允许参与通信的末端节点相互分离，因此节点能够动态地加入和离开分布式应用。DDS 以数据为中心，所有的服务质量（QoS）参数在每一个末端节点的基础上都可以改变。末端的可配置性是支持复杂数据通信模型的关键。

简而言之，DDS 就是按“哪里需要，何时需要”来分发数据。发布/订阅模型使“哪里”变得容易，以数据为中心则解决“何时”的问题。

本章介绍了中间件和通用通信模型的基本概念，并介绍了 DDS 的功能集如何解决实时系统的需求。

1.1 体系集成需求

随着实时分布式系统复杂程度的不断增加和研发规模的迅速扩大，系统集成的难度和风险都在大幅度提高。不同阶段独立研发的各大系统之间需要集成，我们通常称之为“体系”。

2006 年 11 月 2 日，美国海军有意炸沉“福吉谷”号宙斯盾巡洋舰，这艘只服务了 18 年的第四代巡洋舰的设计寿命至少为 30 年，那究竟发生了什么使得美军要主动销毁它呢？这艘巡洋舰的船体设计是合理的，发动机和主要设备也能够正常工作，问题只是在于系统软件无法升级到新的技术和现代武器系统。这是一个令人震惊的事实：系统软件升级的代价已经超过了保留 10 亿美元的资产。为什么会这样？众所周知，复杂系统的集成是十分困难的，协调不同子系统的开发团队更是一项挑战。在过去，唯一的办法就是把整个系统的开发采用“烟囱式（stovepiping）”垂直管理和运作模式。这种模式成本巨大，更糟的是它不鼓励可扩展的设计，导致集成和维护的成本直线上升。从“福吉谷”号这个事件中可以看出，集成和维护的成本超过了船体本身，而这并不是偶然的特例。在很多行业中，实时系统集成的成本无法控制。这种状况再也不能继续下去了！

1.1.1 系统集成的问题

系统集成的成本和风险随着系统和团队规模的扩大不断增加。小型团队不需要太多的协调工作，例如一个团队只有 10 个程序员，他们只需每周开展例会，审查进展情况、检查接口和设计的变更等。100 个人的团队则必须分成若干个小组，文件设计、审查和项目管理逐渐变得很重要。随着团队规模的增加，公司面临的压力是争取并完成更大的项目。该项目可能包括多个公司和组织之间的协作。在这种情况下，接口便成为谈判的细节，技术选择也变得十分困难，整个团队的设计审查成为关键。系统集成的成本大大增加，尤其是前期工作的开销。细小的变动也需要细致的文件备份和进度控制，甚至追加费用。

如果还要考虑时间因素，则问题将变得更为复杂。大型系统的开发和维护通常需要跨越多个时间段，这意味着旧的技术和设计必须以某种方式保留与新版本的良好接口，否则可能带来严重的后果。企图开发无须升级的“定制”软件（类似开源的诱惑）是错误的想法，如果原有软件无法更新到最新的平台，所有相关的技术都必须改变。“重用”是一个永远难以实现的梦想。与升级不同，承包商通过看似简单的更新收取了数亿美元的费用。简而言之，大型系统集成的成本极高并对系统维护有着极大的影响。

如图 1-1 所示，集成规模随着系统的增大迅速膨胀，构建和维护大系统的成本和风险完全由集成主导。

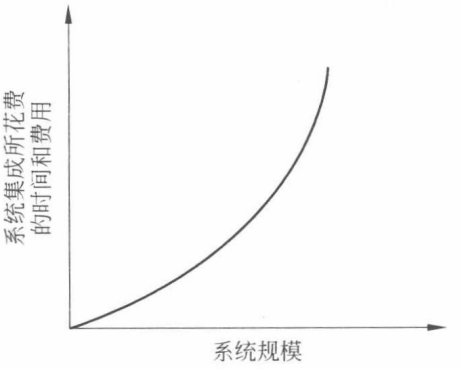

图 1-1 集成成本随系统增大

1.1.2 耦合性问题

系统集成为何如此重要？根本原因在于系统是由很多模块组成的，而这些模块由不同的小组设计。当一个团队变大时，协调和交流就变得十分困难。大型团队构建的系统只能由独立设计、实施和管理的子系统组成，显然这并不容易。我们的第一个挑战是用独立设计的模块构建高性能的实时分布式系统。图 1-2 显示了典型的传统设计模式，即运用“客户端/服务器”或“远程对象”技术，例如 CORBA、ODBC、RMA、OPC 等。在开发过程中，首先必须确定系统功能，然后再设计每一条信息的访问方法和接口。这些数据“隐藏”在各个对象背后，每个设备或端点数据处理的要求事后进行。这种设计假定网络是相对静态的，服务器总是存在和可访问，服务器/客户端关系明确，客户知道在何处，更重要的是何时申请数据，所有应用都有类似的交付要求。它还假定一个“同步”处理的模式，客户端请求服务器，然后等待答复。但随着系统复杂度的增加，这个以“服务器为中心”的设计很容易崩溃，而且这种设计使得增加新的系统或数据（图 1-2 中的虚线）非常困难。每个服务器通常还和各自的客户端耦合，除非现有的接口能够提供所需的一切信息，否则每个与新的

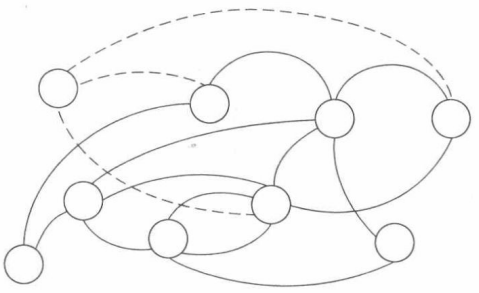

图 1-2 传统设计的紧耦合网络

数据有关系的接口都必须重新设计。

传统技术需要开发人员定义为每个函数访问方法，然后实现服务器和其他特殊接口，这导致“意大利面条”式的网络设计和“烟囱式”的系统。

图 1-3 描述的是一种以数据为中心的设计方法。这种设计方法仅需要开发人员定义各个子系统数据的输入和输出，这与前面隐藏数据的方法正好相反，这种方式中的信息流影响每个模块。

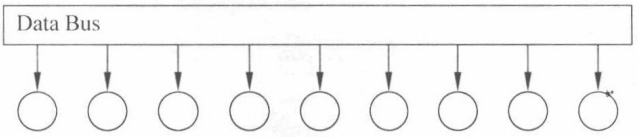

图 1-3　以数据为中心的设计引入了虚拟“总线”的概念

这种中间件能够自动发现信息发布者和接收者，并即时提供所需要的数据。这种“以数据为中心的发布/订阅”设计模式消除了耦合，大大简化了苛刻或复杂系统对数据共享需求的集成难度。因此，对象管理组织(OMG)迅速制定了数据分发服务(DDS)标准，并很快在很多系统中得到了广泛的应用。在全球，大部分军事系统的集成技术都采用了 DDS。它还在金融贸易、自动驾驶辅助系统、空中交通控制和能源等行业得到了广泛和成功的应用。

以数据为中心的发布/订阅模式清晰地定义了接口和模型信息。有了清晰的接口，各个部件很容易接入“数据总线”。可见，DDS 使得模块之间成为松耦合的方式，这使得集成各个系统成为可能，但这还远远不够，DDS 在分布式的指挥控制系统中处理复杂数据流业务具有卓越的能力。

1.1.3　复杂数据流问题

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的各参与者之间采用一种具有松散耦合特性和通信服务质量保障策略支持的灵活通信模型和交互机制。

DDS 在处理复杂的数据流方面做得很优秀，通过对服务质量策略的控制，可以把对更新速率、可靠性和带宽控制有不同要求的模块很好地集成到同一个系统中。DDS 能很好地处理不同种类的数据传输，可包括快速连接的节点、慢速连接的节点以及间歇性的连接(例如无线连接)，这对于构建综合性系统无疑是一个有利条件。

在基于信息系统的体系作战中，体系要处理各种复杂的数据流，这些数据流来自不同的传感器和终端，例如战场通信系统、舰船控制系统和航天测控系统等领域(如图 1-4 所示)。这些复杂的数据流既有传统的应用模式，也有企业级应用模式，还有实时应用、嵌入式应用。当前较为流行的实时 CORBA 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，并且它没有服务质量策略支持，不能满足通信灵活性要求。Java 消息服务 JMS 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，但是它缺乏应用级服务质量策略支持，仍然不适合处理复杂的数据流，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

系统所需的服务质量策略来满足分布式实时应用需求的网络中间件。

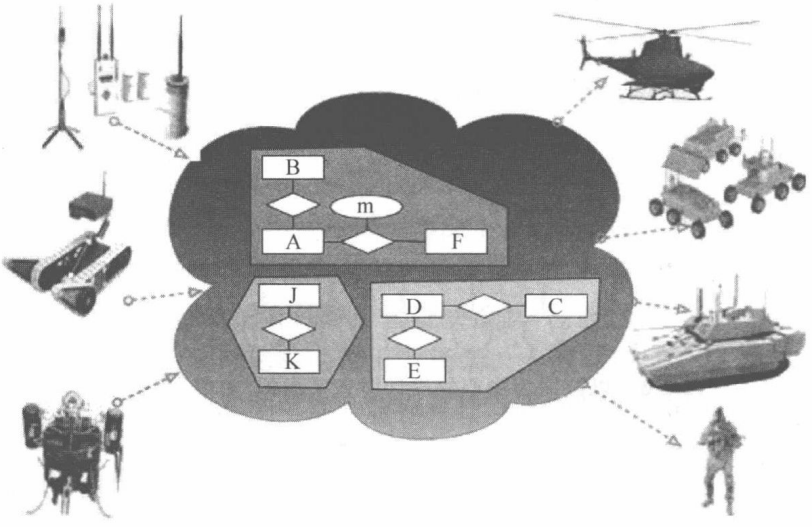

图 1-4 复杂数据流示意图

实时系统网络中间件技术在国外已经非常普及，美国和欧洲各国从 20 世纪 90 年代末就开始应用该技术，在今天，美国的海、陆、空三军各项目中，DDS 已经成为分布式系统中强制性的标准。

1.2 什么是数据分发服务

数据分发服务是一个用于实时分布式应用程序的网络中间件，它提供了程序员在嵌入式或企业设备或者节点之间分发时间关键数据时所需的通信服务。DDS 使用发布/订阅通信模型，使数据分发变得有效和稳健。

很多分布式应用现在就有，其更多的应用在未来。所有的分布式应用有一个共同条件，即必须在不同的执行线程之间传递数据。这些线程有可能在同一个处理器中，也可能散布在交叉的不同节点上。相应地，开发者必须设置多个节点，每一个节点上有多条线程或进程。

这些节点或进程的联系是靠一种传输机制运行的，例如以太网、共享内存、虚拟机总线基架或无线网络。节点之间的标准化通信路线通常用 TCP/IP 这类基础协议或 HTTP 这类较高级的协议。共享内存访问典型的用法是用于同一个节点的进程运行，它也可用于无论何处的公共存储器访问。

图 1-5 所示的是一个简单的分布式应用的例子：一个单片机连接了一个温度传感器电路和以太网，传感器以一个确定的频率采集温度数据。图中的网络中还连接了一个工作站，它是用来给操作员显示数据的。DDS 的实时系统数据分发服务机制使这种数据通信路径更便利。

使用数据分发服务，系统设计员和程序员从灵活的通信基础设施开始，可在广泛的计算机硬件、操作系统、语言和网络传输协议条件下工作。DDS 是高度可配置的中间件，这样程序员能够调整它，使其满足应用程序的特定的通信要求。

总之，使用数据分发服务，用户可以：

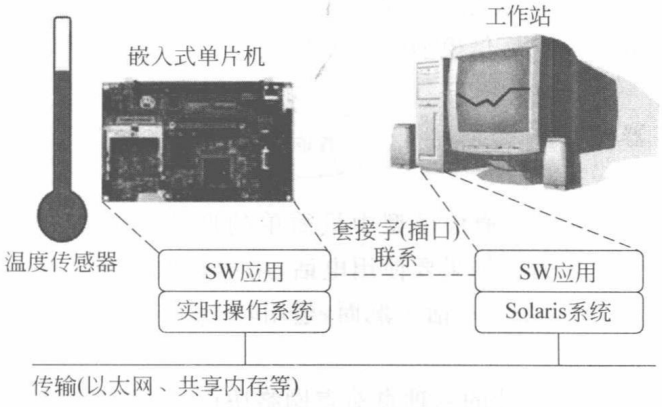

图 1-5　简单的分布式应用

- 执行复杂的一对多和多对多网络通信，应用程序使用 OMG 标准 API、DDS 来发布和订阅数据。
- 自定义应用程序操作，满足不同的实时性、可靠性和服务质量目标。
- 提供应用程序透明的容错功能和应用程序的健壮性。
- 使用多种传输方式。

1.3　什么是中间件

中间件是位于应用程序和操作系统之间的一个软件层。网络中间件将应用程序从基础计算机架构、操作系统和网络堆栈(如图 1-6 所示)的细节中隔离开。网络中间件通过允许应用程序发送和接收信息，简化了分布式系统的开发程序，无须使用低层协议编程(如 SOCKET 或 TCP/IP)。

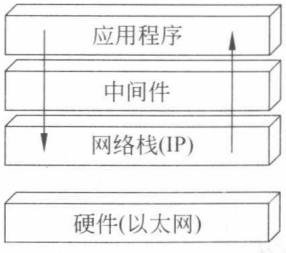

图 1-6　网络中间件

DDS 是基于发布/订阅模式的通信模型。发布/订阅中间件提供一种简单、直观的方式分发数据，它将创建和发送数据(数据发布者 Publisher)的软件与接收和使用数据(数据订阅者 Subscriber)的软件分离开。Publisher 简单声明其发送意图并发布数据。Subscriber 声明其接收意图，然后中间件自动传送数据。

除了模型的简便性，发布/订阅中间件可以处理复杂的信息流模式。发布/订阅中间件的使用将实现更简单、更模块化的分布式应用程序。更重要的是，发布/订阅中间件可以自动处理所有的网络琐事，包括连接、失败和网络变化，这消除了用户应用程序为那些特殊情况进行编程的需要。经验丰富的网络中间件开发人员都知道，处理这些特殊需求要花费 80%的精力和代码。

1.4　网络通信模型

位于网络中间件底层的通信模型是应用程序如何通信的最重要的因素。通信模型影响性能，完成不同通信传输的易用性、检测错误的能力、不同错误条件下的健壮性。不幸的是，

没有“一刀切”的做法来处理分布式应用程序。不同的通信模型适合不同类别的应用领域。下面简要描述 3 个主要的网络通信模型：

- 点对点模型；
- 客户端/服务器模型；
- 发布/订阅模型。

(1) 点对点模型。点对点是通信模型中最简单的形式，如图 1-7 所示。电话是日常生活中点对点通信模式的一个例。如果要使用电话，用户必须知道对方的地址(电话号码)，一旦连接建立，用户可以享用高带宽对话。然而，当用户必须多人同时对话时，电话无法工作。电话实际上是一对一的通信。

TCP 是 20 世纪 70 年代设计的一种点对点网络协议。TCP 提供了可靠的高带宽通信，但它难以应对拥有多个通信节点的系统。

(2) 客户端/服务器模型。为了解决点对点模型的可扩展性问题，开发者转向了客户端/服务器模型。客户端/服务器通信模式指定一个特殊的服务器节点，它可以同时连接多个客户端节点，如图 1-8 所示。客户端/服务器是一个“多对一”的架构。通过电话订购比萨，是一个客户端/服务器模型的通信例。客户端必须知道比萨店的电话号码并下订单。比萨店可以处理多个订单，而不必提前知道客户端的位置。下订单之后(请求)，比萨店询问客户端(比萨)应发送至何处。在客户端/服务器模型中，每个响应绑定一个事先请求。结果是，该响应可适合每个请求。换句话说，每个客户端发出请求(订单)，而每个回复(比萨)用于一个特定的客户端。

图 1-7 点对点模型

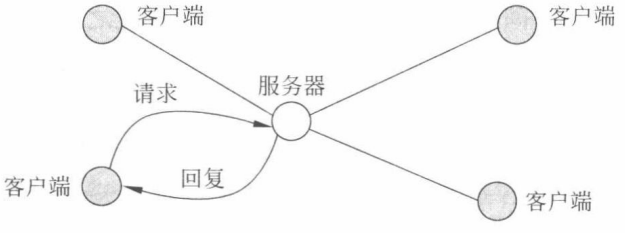

图 1-8 客户端/服务器模型

当信息集中的时候，例如数据库、事务处理系统和文件服务器，客户端/服务器网络架构的效果最好。然而，如果多个节点生成信息，客户端/服务器架构要求所有信息发送到服务器，随后再分发到各客户端。这种方法低效，并排除确定性通信，因为客户端并不知道新信息在服务器上何时可用，而且当客户端请求和接收数据时，它为系统埋下了不确定的种子。

(3) 发布/订阅模型。发布/订阅应用程序是一种典型的分布式应用程序，它们之间通过终端节点匿名发送(发布)和接收(订阅)数据，如图 1-9 所示。通常，若发布者要与订阅者建立通信，只需知道数据的名称和定义就可以了。发布者和订阅者都无须知道对方的其他信息。只要相关的应用程序知道它正在传输何种数据就能够通过发布/订阅模式将数据传递至恰当的节点，并且不用单独建立连接。发布者负责收集恰当的数据并将数据发送至所有注册的订阅者。订阅者负责从发布者那里接收恰当的数据并将数据呈现给感兴趣的用户。

在发布/订阅通信模型中，计算机应用程序(节点)“订阅”它们所需要的数据，“发布”它