

82-84/81

视频点播系统的开放式分布处理视点^{*}

ODP Viewpoint of VOD System

何炎祥 黄浩 张戈 石莉 李超

(武汉大学计算机科学系 软件工程国家重点实验室 武汉430072)

Abstract VOD(Video On-Demand) is one of the most important application of distributed multimedia system. The design of the experimental VOD system is discussed using the approach, Open Distributed Processing(ODP), based on multiple viewpoints.

Keywords Open distributed processing, Video On-Demand, Reference model-ODP, Viewpoint

1 引言

分布式多媒体系统是计算机网络技术、通信技术、多媒体信息处理技术、分布计算技术、数据压缩技术等众多技术相互结合的产物,其应用已涉及相当多的领域,如教育、办公、新闻、出版、医疗、金融、计算机支持协同工作(CSCW)等。分布式多媒体系统中一个十分重要的应用是视频点播(VOD),它将远程视频服务器上的媒体信息,通过计算机网络或其它通信设施传送到本地的媒体处理器和媒体终点上,本文利用 ODP(开放式分布处理)方法^[1]讨论了 VOD 实验系统的设计。

ODP 是超出 OSI(开放式系统互联)^[2]范围的一个崭新框架。在实践中,人们逐渐认识到,OSI 中许多面向通讯活动的标准化其实需要一个比 OSI 参考模型更广阔的框架,ODP 是为能够在异构且多厂家环境中构造分布式系统而提供的一个所有分布式系统都遵守的通用的体系结构框架;这样一个框架不仅包括分布式计算的标准,而且还包括在开发系统中进行分布式计算所应遵循的开发标准,需要说明的是,ODP 是基于视点(viewpoint)技术^[3]的。

本文简单介绍 ODP 参考模型和 ODP 视点,然后在文[4~7]的基础上,借助 ODP 参考模型的五个视点,对 VOD 实验系统进行描述和设计,最后,讨论了 ODP 方法的一些技术难点。

2 开放式分布处理(ODP)

2.1 ODP 参考模型

ODP 参考模型(RM-ODP)^[1,6,9]的应用范围十分

广泛,它支持所有传统的数据处理系统、网络化的个人计算机、实时系统和多媒体系统。单机其实可以看成是分布式系统的特殊情况,因为通过扩展或与其它系统连接等方法,可以将原来独立的系统最终转变为分布式系统,所以可以将所有系统都看成潜在的分布式系统,这样,ODP 参考模型通过强调分布的潜在性,实际上为所有系统的功能说明提供了一个框架,同时,自然使得其设计和实现所用的投资能产生最大的效益。

ODP 参考模型由四个部分组成:第一部分对 ODP 参考模型进行了介绍,并且举出一些 ODP 系统的例子;第二部分集中介绍了一些 RM-ODP 中模型化用到的概念,这些概念是描述分布式系统和分布式系统所支持的各种企业所必需的,其中采用的方法是基于对象的方法;第三部分介绍了更加细化的概念,以及一个系统成为 ODP 系统所必须遵守的规则;第四部分给出了结构化语义(architectural semantics),并用一些形式化描述技术(FDT),如 LOTOS,SDL,Z,Estelle 等,解释了第二部分和第三部分中的基本概念。

分布式系统的需求分析和说明是一个十分复杂的工作:只要不是很小的分布式系统,其完整的说明必须包含很大的信息量,因此,试图用一种描述就把握设计的方方面面是不可能的,运用多视点(multi-viewpoints)技术是一个可行的办法,多视点技术将一个复杂的系统设计问题进行适当的分解,使之成为一个易于管理的形式,ODP 参考模型提出五个视点:企业视点,信息视点,计算视点,工程视点和技术视点。参与者可以由其中任意一个视点出发得到整个 ODP 系统的需求说明。这样,使得在设计阶段中能够将关注不同领域的系统约束条件加以分离,同时,视点也使得不同的

^{*} 国家教育部重点项目资助课题

参与者都能从一个恰当的角度观察这个分布式系统。

2.2 ODP 视点

·企业视点 所关注的是处于一个组织中的分布式系统的职能和用途。(这里,使用“企业”这个词不仅限于单个组织,其实,构造出的模型需要描述出若干不同组织间在相互作用上的约束条件)。具体地说,企业视点主要说明的是待说明系统的商业用途,即这个分布式系统是做什么的,以及什么人出于什么目的在什么策略下使用这个系统,其中包括社会策略、管理策略、经济策略和法律策略;同时,企业视点系统结构和参与者的职责之间的关系,参与者的行为和目的之间的关系也进行了描述。企业视点给出的模型说明了系统和与其交互的环境。

通常,企业的管理层从企业视点观察和说明分布式处理系统。

·信息视点 所关注的是分布式系统所处理和保存的信息。信息视点包括两个方面:一是要对信息元素进行说明,即要说明信息元素的结构、关系、特性和逻辑划分等;二是要对信息进行建模,说明信息在什么地方生成,如何流过整个系统,又在什么地方进行处理和保存。信息视点通过将信息模型分解到各单个部件中,得到一个一致的公共视图。在说明信息发送方、信息接收方、和两者间信息流时,就会用到这个视图。

通常,商业分析员和系统分析员从信息视点观察和说明分布式处理系统。

·计算视点 所关注的是作为信息发送方和信息接收方的软件部件的结构和它们相互之间的关系,计算视点将分布式系统功能分解为对象的集合;对象在接口的交互关系使得系统的分布成为可能。这样得到的用于支持分布式的逻辑模型反映出系统的功能需求,而且,这个模型并不依赖分布式系统的物理环境,也就是说,它独立于底层网络和计算机的实际配置。

通常,系统分析员和应用设计者从计算视点观察和说明分布式处理系统。

·工程视点 所关注的是系统用以支持计算模型和支持分布式所必须的底层机制和各种透明度。底层机制从逻辑上将计算对象和底层技术分离,从而使得计算对象可以根据特定的应用需求实现分布。工程视点说明了计算对象如何通过底层的一些功能组织在一起,以及计算对象如何与底层进行交互;在透明度的问题上,工程视点在分布式物理部件的基础上提出实现这些透明度的方法。

通常,系统编程人员从工程视点观察和说明分布式处理系统。

·技术视点 所关注的是构造分布式系统的部件细节,包括物理硬件和软件部件。另外,技术视点也描

述了体系结构中的概念在实际中是如何实现的,因此,需要考虑所用技术产生的约束条件。

通常,系统编程人员、系统操作人员和系统维护人员从技术视点观察和说明分布式处理系统。

从某种意义上说,ODP 参考模型的确反映出—个典型的设计过程的各个阶段,从开始分析一个企业到最终实现。ODP 的五个视点作为同一系统的不同抽象,不应该看成层次结构,而且,五个视点的次序并不意味着任何特别的优先关系。图1说明了五个视点之间的关系。从图1可以看出,没有一个视点比其它视点更底层或更重要。一方面,各个视点彼此独立,是整个系统说明的部分视图;另一方面,同其它视点模型一样,各个视点不是完全独立的,而是相互联系,它们在一起形成对一个分布式系统的完整分析。

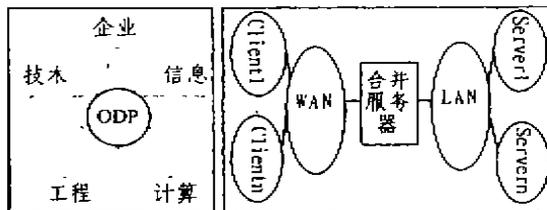


图1 ODP 视点之间的关系 图2 VOD 的体系结构

3 视频点播系统的开放式分布处理视点

下面,我们用上述的开放式分布处理视点来描述视频点播实验系统。

3.1 企业视点

VOD 系统由五种不同部件组成:服务器,服务器端网络(LAN)、合并服务器、客户端网络(WAN)和客户。图2说明了VOD的体系结构。服务器连接到LAN上,是视频服务器的存储体,所有的视频数据都存储在服务器上。服务器不是直接和客户通信,而是通过LAN接受合并服务器发送的start/abort服务的命令。服务器端网络用来连接服务器和合并服务器。合并服务器作为位于WAN和LAN之间的设备,在整个VOD系统中占据一个重要的地位。它有两个功能:将多个服务器送来的视频流合并成为一个视频流送往客户;对预取的数据进行缓冲。客户端网络连接着合并服务器和客户。客户是视频数据的最终消费者。它可以是个人计算机、工作站或有线电视顶盒。

3.2 信息视点

信息视点使用模式这个概念对VOD系统的信息对象和信息间交互关系进行描述,VOD系统的信息视点要定义出视频流模式,主要说明了视频流的MEPG编码格式、服务质量(QoS)、视频流的管理等。信息对

象的模式分为：固定模式、动态模式和静态模式。例如，VOD系统的说明可以使用固定模式，定义出一个阈值来控制视频流所必须达到的吞吐量，进而保证服务质量。

3.3 计算视点

计算视点将VOD系统看成是计算对象的集合。对于每个对象的接口，ODP参考模型定义了三种不同类型：操作接口、流接口和信号接口。图3中，计算视点把VOD系统中的服务器、合并服务器和客户作为具有三个接口的对象：控制接口（属于操作类型接口），操作接口和流接口。控制接口用来支持VOD服务控制和管理的操作；操作接口用来支持VOD服务所提供的功能；流接口用来支持服务器和合并服务器之间，合并服务器和客户之间的视频流。

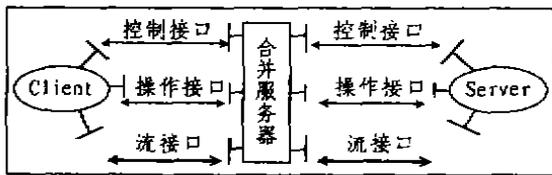


图3 VOD系统的计算视点

3.4 工程视点

工程视点描述VOD系统的企业视点、信息视点和计算视点三者的工程化实现。图4表明计算视点中的计算对象是如何映射到工程视点中的工程对象，注意，并不存在一个从计算对象到工程对象的一一映射。在客户、合并服务器和服务器三者之间建立一个通道；有三种不同的通道，分别用来传输控制接口和操作接口上的操作以及流接口上的视频流。Stub对象、Binder对象、Protocol对象在整个通信活动中有各自不同的作用。Stub对象提供用以支持分布式透明度（如，从一种视频信号编码到另一种视频信息编码的数据格式转换）的适应性功能。Binder对象确保相连的接口之间的兼容，维护流绑定的完整，管理配置和出错处理。Protocol对象提供通信功能，如传输协议，用以支持多个Protocol对象间的信息传输；Protocol对象的选择要符合QoS参数的限制。

3.5 技术视点

技术视点将VOD系统的说明和VOD系统的实现联系起来，从而将VOD系统的说明和现实世界联系起来。例如，技术视点描述了VOD系统对高性能工作站的需求，VOD系统需要它保存、传输和显示视频信息；同时，为了支持若干需同时传送的视频流，VOD系统需要服务器端网络应是高带宽低延迟的网络，对

于客户端网络而言，与服务器端网络不同，VOD系统并不需要它对每个客户都提供很高的带宽，但对某些客户，它需要提供足够的带宽。

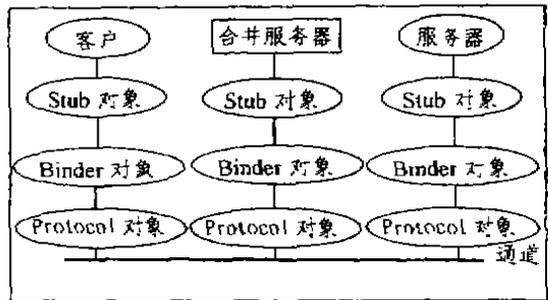


图4 VOD系统的工程视点

4 讨论

4.1 视点技术

能够将注意力集中是视点技术最大的好处。实践表明，ODP的五个视点易于理解，而且对于典型的软件系统分析和设计过程而言，它们是完备的。虽然每个透视或视图是对整个分布式系统部分且不完整的描述，但是视点技术允许不同的主体从不同的视点收集系统信息，因此，丢失系统信息，尤其是系统关键信息的可能性已经大大地减小，并且它把一个复杂、难于管理的任务适当地分解成为若干简单、易于管理的子任务。

4.2 视点间的一致性

引入多视点方法到系统说明后，对同一实体或相关实体的描述就有可能出现在不同视点中，因为彼此需要共存，所以视点间的一致性就成为一个十分重要的问题；ODP的五个视点产生的五个同一系统的描述需要相互保持一致。例如，编程人员的工程视点就有可能与技术视点中的硬件和软件配置发生矛盾。因此，虽然ODP不是一个设计方法学，但它要求系统的说明是精确不模糊的解释。图表虽然可以给出一个概况，但图表的描述太粗糙，无法给出编程人员所需的细节。因此，形式化的方法在ODP中处于很显著的地位。

4.3 形式化描述技术

“RM-ODP的贡献在于尽可能地使用形式化描述技术(FDT)，使得能够对RM-ODP作出清晰不含糊的说明。”^[1]在ODP参考模型中，每个视点都有与之对应的视点语言，用以说明此视点关注域的规则。ODP参考模型第四部分中结构化语义的提出，使得视点说明可以用若干形式化描述技术来书写，如信息视点可以

(下转第81页)

同时参与了两个多点播送操作 Y 和 Z,那么一个并发操作的算法应该避免操作 Y 和 Z 在同一步要求 X 发送消息,否则将导致结点争用,操作 Y 和 Z 将在 X 结点上串行,从而延迟两个操作的时间,即使 Y 和 Z 的参与者是不相交的,各自的参与者也可能在并行系统的同一块内,从而导致链路争用。

此外,其他的通信操作如路障同步、收集、全局归约都受到类似问题的困扰,因此,任何为多个并发操作而设计的算法需要清楚地定义和评估自己的适用范围。

3.3 算法观点

现在,让我们从算法本身的角度来分析实用性问题。

3.3.1 同步与异步 许多算法都采用同步的方法来解决如全对全通信等^[9,10,17]问题。也就是说,所有的参与者同时开始一个通信步,也同时结束该步,在每一步内,是没有资源争用的。这种策略使算法可以很容易得到通信复杂度的上下界。然而,由于内在的同步性,当其在目前主流的 MIMD 机器上实现时,它们的性能并非如所期望的那么好,这使得在实际应用中,很难选择合适的算法。因此,在设计算法时就应该对机器的内在异步性加以考虑。

3.3.2 直接、间接与混合方法 一些为全对全通信而设计的算法在本质上就是直接型的,每个结点通过不同的通信步将消息送往每一个目的结点。在间接型的算法^[9,12,18]中,结点可以从多个结点接收消息,排列数据,重组消息,然后将它送往另一个结点集。还有一些算法组合了直接和间接两种方式,称为混合方法。除了通信启动时间和消息传播代价外,间接型和混合型算法的性能还受到数据排列转换代价的影响。这种

代价还取决于数据是从缓冲区直接传入缓冲区,还是从缓冲区到存储器再到缓冲区。虽然现在的算法考虑了数据转换代价,但缓冲区的数目和用途需要清楚地说明。另一个重要的问题是绝大多数间接型和混合型算法忽略了系统所允许的最大消息长度。这些算法的最优性能是在消息长度不受限的情况下得出的。为了使这类算法实用,在性能分析时应该考虑最大消息量^[13]的限制。

3.3.3 线性时间模型与常量时间模型 现在,我们考虑可能的通信时间模型。大量的实验表明,两个结点的通信时间极大地依赖于消息长度^[11,13],线性时间模型遵从这一点,一个通信步所需的时间随着消息量的上升而成比例上升。遗憾的是,很多算法为了理论证明的方便,简单地采用了常量时间模型^[13],在这种模型里,一个通信步只需一个时间单位,无论消息的量有多大,传输路径有多长;而且消息的分裂和重组被假定为不需要时间。因此采用常量时间模型的算法通常“具有”更好的性能。在实际应用中,这样的模型几乎不可能在蛀孔算法(wormhole)之外的机器上实现。由于采用常量时间模型的算法难以给出真正的时间复杂度,因此无法与采用线性时间模型的算法进行比较。为了让这类算法变得实用,应该将消息量和路径长度考虑在内。

结论 本文从系统、应用和算法三个角度对并行通信算法的实用性进行了分析研究,探讨了如下几个方面的问题:1)如何确定现存算法的应用范围;2)如何正确全面地分析算法的各种性能;3)如何为当前和未来并行系统设计更实用的算法。同时,本文也给出了一些相应的解决措施。

(参考文献共19篇,略)

(上接第84页)

用 Z 语言来说明,计算视点可以用 LOTOS 语言来说明。可以看出,特定的规范说明语言可以作为特定的视点语言,而且每个语言的描述程度是不同的。这就引出一些问题:对于每个 ODP 视点选择何种 FDT 最好;如何保证不同 FDT 书写的不同视点说明之间的一致性等等。

基于视点技术的开放式分布处理是设计分布式多媒体的一个比较好的方法,但现在还有一些亟待解决的问题。目前已有一些学者开始着手解决这些问题。

参考文献

- 1 ITU Recommendation X. 901-904—ISO/IEC 10746 1-4, "Open Distributed Processing—Reference Model—Parts 1-4", July 1995
- 2 Zimmermann, H. OSI-reference model—ISO model of archi-

- 3 tecture for open system interconnection. IEEE Trans., 1980, COM-28: 425~432
- 4 Editorial: Viewpoints in Requirements Engineering. Softw. Eng. J., 1996, 11(1): 2~4
- 5 Berzsenyi M, et al. Design and implementation of a video on-demand system. Computer Networks and ISDN systems, 1998, 30(16/18): 1467~1473
- 6 Leydekkers P, Gay V. Multimedia conferencing services in an open distributed environment. IWACA / 94
- 7 Geijs K, Mann A. ODP viewpoints of IBCN service management. Computer Communications, 1993, 16(11): 695~705
- 8 Bowman H, et al. Cross-viewpoint consistency in open distributed processing. Softw. Eng. J., 1996, 11(1): 44~57
- 9 Linnington P F. RM-ODP the architecture. In: Raymond K, Armstrong L, eds. IFIP TC6 Int. Conf. On Open Distributed Processing. Brisbane, Australia, February 1995. 15~23
- 10 Geijs K. The Road to Open Distributed Processing (ODP). In: Proc. GI/NTG Conf. "Communication in Distributed Systems. Mannheim, Germany, February 1991