

# 开放系统中的实时性

Real-Time Problems in Open System

骆志刚 胡 健 刘锦德

(电子科技大学计算机学院 成都610054)

**Abstract** With the wide spread application of open system technology and the new trend of transition from proprietary real-time system to open real-time system, the demand for careful study on real time problems for open system is emerging. In terms of the features of real-time system and the characteristics of open system environment, it is deduced that open real-time system contains two aspects: communication real time and end-system real time, and is suggested to use real-time domain concept to divide system into smaller subsystems. Real-time QoS is introduced to express various real-time requirements. Design and implementation of the open system real time are discussed.

**Keywords** Open system, Real time, Real-time system, Real-time domain, Real-time QoS

## 1 前言

开放系统技术在信息技术领域已经得到了广泛应用,应用开放系统技术可以方便地开发、集成、升级和维护各种应用系统,大大降低了构建应用系统的代价,并极大地提高了生产效率。尽管开放系统不存在精确的定义,但对开放系统却存在着公认的“必须具备特征”<sup>[1]</sup>,也即它们是:具有可移植性;具有可互操作性;具有可伸缩性;具有易获得性。

由于开放系统短暂的历史(计算机领域出现开放系统的名称只有十多年的历史)和多变的有关技术环境(用户的用机条件从单机变为局域网,又从局域网变为广域网;应用从简单的单机应用变为网上应用,特别是网上多媒体应用),使开放系统的内涵迅速地变化和丰富。以下两种发展趋势促使我们有必要对开放系统中的实时性进行深入的探讨。

(1)普通的开放分布式计算环境向实时系统发展

由于通信技术和计算机技术的迅速发展,带宽和CPU速度的不断提高,语音和视频的实时处理成为可能。许多原为非实时的应用要求得到扩展,以便进行实时语音和实时视频的处理。支持具有时限操作的需求,出现在不同于传统实时系统的系统中。具有家用娱乐系统的家用计算机,需要能够保证网上数字视频和音频的质量,实时需求也已经开始进入传统的办公计算机系统,如通过网络环境,进行视频会议、网上教学等多媒体应用。这些趋势要求实时功能成为系统的固有性质,这样就要求普通的开放分布式计算环境将实时功能作为普通服务来提供。

2)实时系统从专用实时系统向开放实时系统发展<sup>[2]</sup> 实时应用的规模不断扩大,大规模分布式实时系统日益广泛,电信、交通、作战等信息系统都是常见的大规模分布式实时系统。这些庞大的实时系统除了面临实时运行的问题外,同时也面临其它任何大规模软件系统所遭遇的问题,如开发困难、维护困难等。传统的实时系统必须经过精心定制以便能在专用系统上运行。虽然这些定制的系统为实时应用的运行提供了所需条件,但是它创建了一个不灵活的结构,存在不容易修改、升级和与第三方产品集成的缺点。用这样的方法来完成如此庞大的实时系统的开发、集成和维护,其代价极大。唯一的解决办法是使用开放系统技术,发展实时软件工程以解决大规模实时系统所面临的问题。基于COTS(Commercial-off-the-shelf)的实时系统就是开放系统技术在实时领域中的应用实例<sup>[3]</sup>。

一个具有实时性的开放系统环境可以同时包含实时对象和非实时对象。为非实时对象提供的各种操作和机制可以应用于实时对象,例如多种分布式系统机制(如交易、安全、定位、联合等)可以应用于实时对象。这样一个系统不仅可以提供包含实时功能在内的多种功能,而且由于它采用开放系统技术,容易构建和维护。

## 2 实时系统的特征

从揭示“时间”的重要性出发,实时系统可以描述为:在实时系统中,时间是一种重要的资源,对外部事件的响应和任务执行都必须在限定的时间内完成。在分布式系统中,还必须在限制的时间内完成消息的发

送和接收,实时系统中,输出结果的正确性不仅取决于计算所形成的逻辑结果,还要取决于结果产生的时间,一个实时系统,具有以下几个重要特征:

1) 可预测性:是指系统所执行的操作按预先定义或确定的方式执行,其操作执行的时间是可预知的。这一点是实时系统最重要的特征,可预测性将应用于实时系统环境的每一个组成部分,这个环境才能提供一定程度的可预测性,根据实时应用的要求,实时系统环境提供不同程度的可预测性。

2) 及时性:实时应用不同于非实时应用在于它们的操作具有严格时限,实时系统中,输出结果的正确性不仅取决于计算所形成的逻辑结果,还要取决于结果产生的时间,通常实时环境中,按照实时活动的最后期限(deadline)和当前可用的资源来进行调度,以使实时操作能在最后期限到达前完成。

3) 用户控制:用户控制表示用户对系统的行为具有有效的控制能力。许多实时应用是嵌入式系统,这样易于实现对于系统行为的控制,另一方面,实时应用的行为具有多样性,固定的系统行为不能满足多个实时应用的需求,这就要求用户具有强有力的控制能力。最简单的系统行为的控制方法是为实时活动选择适当的优先级,从而控制应用的响应时间,除此外,可以选择调度策略、资源分配策略等,与非实时系统比较,实时系统需要能从较低层次进行更多的干预。

4) 任务定向:任务定向表示整个分布式系统通过一个或多个分布在不同节点上的应用程序相互合作,专用于完成一个特定的目标。从实时意义上讲,任务定向表示任务的成功程度依赖于整个系统所获得的与实时约束有关的信息。在其最简单的形式中,当一个任务跨网时,任务相关的优先级或最后期限具有全局意义。全局重要性和紧急程度的特征能在系统中传递,以解决整个系统资源竞争问题。

5) 性能优化:实时系统通常具有较高的性能要求,性能优化在实时系统中更为重要。实时系统需要很好地处理功能、粒度和灵活性的关系,以达到最优的性能。

除以上特征外,实时系统还具有并发性、可靠性、可嵌入性等特征。

### 3 开放系统中的实时性

传统的实时系统通常是专用的、封闭的系统,系统包含的应用可以预先确定,其实时性的满足是通过全局分析来完成的。系统的开放导致应用无法预先确定,这就对实时的保证提出了新的挑战。

#### 3.1 开放系统环境的特点

在开放系统环境中,往往存在着许多不同的节点、

资源和应用,地理上分布着的节点可互联、互通和互操作,以实现应用的合作处理和信息的共享互用,为用户提供形式多样的应用和服务,随着各个企业事业所用的计算机系统的规模不断发展,越来越多的计算机系统被联一起,表现为规模庞大的开放式分布结构。一个开放系统具有以下几个特点:

· 异质性:开放系统环境的组成复杂,硬件、软件和技术具有多样性,开放系统要将这些分布在不同平台上、采用不同技术的应用集成在一起,协同工作;

· 联合性:一个开放系统可能跨越多个自主实体,为完成一个目标,不同管理域,技术域需要联合起来;

· 伸缩性:一个开放系统在规模上是可伸缩的;

· 发展性:随时间发展,开放系统环境是发展变化的,在技术上、决策上、管理上会不断更新。

这样的开放系统环境由于节点分散、资源易于动用、系统不断变化、管理者距远莫及等原因,系统难于协调,不能将其看成一个具有特定实时要求的专用系统,也就不可能通过全局分析来保证整个系统的实时性。因此,在开放系统环境中,需要引入新的策略和机制来保证系统的全局实时性。

#### 3.2 开放系统环境中的实时服务

开放系统在逻辑上可抽象为:由域(domain)和互联(association)两者组成,如图1所示。互联在物理上是以数据通信设备和局域网,甚至广域网实现;而域则由端系统(endssystem)组成。

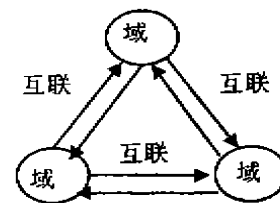


图1 系统抽象

为了确保整个系统的全局实时性,图1中的各个组成部分都应当是实时的,即它们分别采用实时域和实时互联,因此要实现整个系统的实时性,应关注两个问题:端的实时性和通信实时性。因此,开放系统环境中的实时服务应包括两个方面:实时通信服务和实时端系统服务。

1) 实时通信服务:实时应用需要具有严格性能保证的实时通信服务,对延迟、延迟抖动、吞吐率和丢失率有严格的限制,传统的实时应用通常在实时专线(如MIL-STD-1553B、BITBUS、ARCnet等)上运行,以保证通信的实时性。而当前普遍采用的网络结构和协议主要设计来支持尽力型(best effort)服务,对实时应用,特别是硬实时应用的支持,难以满足。随着高速网

络的迅速发展,网络多媒体的应用要求,对网络 QoS 的研究日益深入,特别是网络交换技术的应用,将为通信的实时性提供保证,当前有许多实时应用就是使用 ATM,它将应用的实时要求映射为 ATM 的 QoS,使实时性可以得到保证。

2) 实时端系统服务 端系统的实时性包括了操作系统的实时性和应用的实时性,对于非专用的实时系统(如家用系统和办公系统),操作系统可以选择具有一部分实时特征的通用操作系统,如 Solaris 和 Windows NT,若是专用系统,对时限要求极高(如航空控制系统和武器控制系统)的系统,则应当选择专用的实时操作系统。

开放系统的实时性是全局的、孤立的实时通信和实时端系统并不能保证实时性,只有将两者有机地结合起来,才能确保整个系统的实时性。

### 3.3 实时域

开放系统环境的组成极其复杂,要象专用系统那样通过系统全局分析来保证系统的实时性是不可能的,为此,我们提出实时域的概念,以对系统进行“分而治之”。

实时域是由某特定的实时策略所定义的一个实时环境,其成分有:唯一的实时策略;实时管理机构;实时服务和实时机制,由于在开放分布环境中,各个系统的系统管理员可以分别制定各自的实时策略,因而可能存在许多实时策略和相应的实时域,也就是说,某个实时域内的所有实体都服从该域的实时策略,可使用的实时服务和实时机制类型是一样的,因此,实时域是整个开放分布环境中的用户和资源的一个子集,该子集的元素服从同一个实时策略。

实时域可以通过各种结构互联起来,实时的开放系统环境是由许多实时域和其实时互联所构成的,联接双方既可属于同一个实时域,又可分属于不同的实时域,虽然域内的联接活动与域间的联接活动在逻辑上没有差异,但对于在同一个实时域内的对象,由于实时环境相同,实时联接易于实现;而对于分属于不同实时域的对象,建立联接的问题就复杂多了,只有设法使参与联接的实时域的实时策略兼容时,才能建立联接。

### 3.4 实时 QoS

开放系统环境的特点决定了开放系统中实时要求多样,而且多变,这样多样、多变的实时要求不可能通过固定或静态的方式表达、分析和完成,为此,我们在开发系统环境中引入实时 QoS,来表达、分析和完成开放系统中的实时要求。

实时 QoS 是 QoS 的具体分支,开放系统环境借助实时 QoS 的映射、接纳控制(admission control)、动态协商和适配来满足多样、多变的实时要求,实时 QoS

着重强调以下几个方面

- 时间:如实时活动的最后期限、延迟等;
- 空间:如实时通信所需要的带宽;
- 紧急性:实时活动的紧急程度,服务失败造成的后果;
- 可靠性:失败的概率,恢复的概率等。

实时 QoS 的保证是通过在适当的时间提供足够的资源来完成,因此,在开放系统环境中要能够对资源的分配和调度进行精确的控制。

## 4 开放系统实时性的实现方案

### 4.1 相关技术

图2给出了开放的实时系统可以采用的技术及相互关系,它集中了实时系统技术、开放系统技术和面向对象技术。

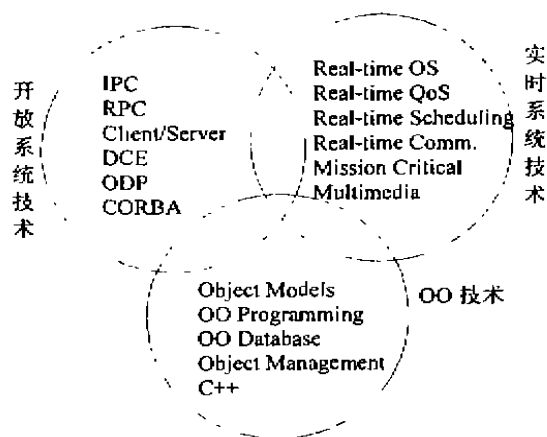


图2 三种技术关系

实时系统提供了各种资源管理,可以加以利用,来保证严格的时限操作。

开放系统技术提供了分布、发展和联合等功能,一个开放分布式系统环境提供了多种分布式透明性机制来支持多个应用成分的透明协同工作,分布式透明性屏蔽了由系统的分布所带来的复杂性,分布式透明性使应用编程者不必关心系统是分布的,还是集中的,从而可以集中精力设计具体的应用,这大大减少了分布式编程的复杂性,另外,应用编程者可以选择自己所需要的透明性,例如:RM-ODP<sup>[4]</sup>定义了八种透明性:访问透明性、位置透明性、迁移透明性、失败透明性、重定位透明性、复制透明性、持久透明性、事务处理透明性。

面向对象技术提供了软件重用和维护功能,实现对象模型化具有多种突出的优点:适用性强、一致性好、模块性好、分类性好和通用性强,可以这样认为,ODP 和 CORBA 是面向对象技术在分布式系统中的

成功应用实例。

实现开放系统的实时性就是要有有机结合以上二种技术,在开放系统环境中引入实时策略和实时机制。这种结合并不容易,例如分布式透明件给用户带来极大好处,但是它是以时间为代价而实现的,这显然与“实时”存在冲突。

#### 4.2 设计原则

对于开放系统环境,其实时性应具有两个主要特点:功能性和适用性。前者指的是这实时系统能提供所需要的实时服务,后者是指它又能适应用户所需的多种的应用形态。

在实际的一个应用中,并不是每时每刻都需要实时服务,仅当在需要的时候才希望提供;而在不需要时,则并不希望提供实时服务,因此,除上述两个主要特点之外,它还应:

- 实时系统的实时性应当是可选用的。

- 从开发效率和开销的角度,开放系统环境中内嵌着基本的实时策略和实时机制,如提供实时 QoS 的表达和执行。而其它的实时服务,如实时调度服务、实时并发服务则可作为增值服务来提供;

- 实时系统是可以动态配置的,许多实时系统在运行过程中,不能随意停下作系统更新,这就要求系统在运行过程中可以动态地配置。

- 既便于用户的使用,又易于操作员的管理;

由此可以导出支持实时的开放系统设计原则是:

- 能支持实时应用和非实时应用的合作;

- 尽可能多地提供所需要的实时服务;

- 容易适用于各种实时系统,从传统的各种实时系统,到新兴的网络多媒体服务;

- 对用户和应用来说,实时系统应是透明的和可选用的;

- 能用于各类应用环境,从单个 PC、主机系统、嵌入式系统、LAN 到开放分布式环境;

- 提供各种管理功能,以便对实时系统进行安装、监测、优化和重建;

- 按组件的设计方法进行设计、构造和建立,以使实时服务可以动态配置。

由于实时系统要完成的功能十分复杂,自然会设想为:实时系统应采用组件化设计的方法,将其复杂性分散到各个功能组件中,使设计与实现简化,便于测试和评价。各个组件是全局实时系统的自治部件,各个自治部件之间相互协作,组成优化的结构,提高操作效率。对全局实时系统的划分要保证各个组件组织起来既要有层次性,又要有结构性。将全局实时系统按层次结构进行组织,下层向上层提供服务,上层为下层隐藏细节和统一差异。

• 60 •

虽然所完成的任务十分复杂,但实时系统必须是一个由有限组件集合而成,且适应性强的系统。这就要求实时系统的基本部件必须具有足够的通用性。这所谓的通用性,就是指在设计 and 实现部件时,应使其只出色地完成有限的特定功能,可方便地自由组合和搭配,这就是为什么在实现实时系统时,要将所需要的实时服务功能由其上层的实时机制来提供,而实时机制建立又借助于其下层的基礎模块来实现。

#### 4.3 实现模型

在一个开放系统环境中实现实时功能,首先面临的问题是平台的异质性,为此,我们在中间件的基础上,来实现开放系统的实时性。

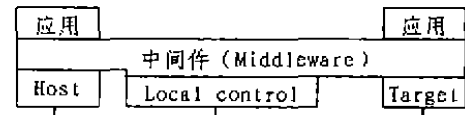


图3 实现模型

图3给出了一个基于中间件的实现模型。当前主要使用一类支撑性软件——中间件来屏蔽平台、网络之间的差异,为应用之间的协同工作奠定基础。如 IONA 的 Orbix, Inprise 的 Visibroker 等已经得到广泛的使用,虽然当前的中间件并不适用于具有严格时限的应用,缺乏对实时的支持,但是它们为实时系统提供了大量的好处:开发简单性、可互操作性、灵活性、可维护性、可重用性等,这些优点都是当前其它实时系统无法提供的。这一类中间件提供了真正的通用软件总线结构,它可以去掉当前实时系统的不灵活结构,因此中间件与实时系统的结合自然成了今后实时系统的一个重要发展方向。基于实时 CORBA 的 TAO<sup>[5]</sup>及非 CORBA 的中间件 ARMADA<sup>[6]</sup>,就是使用中间件技术来构建实时系统。

一般认为中间件不适用于实时应用,且常常受到中间层降低性能或不够快的批评。这一误解的根源在于未能区分“实时”和“快”的差别<sup>[7]</sup>。

为了支持实时应用,中间件应该作以下的扩展:增加实时调度策略和机制;增加实时事件支持以提供对实时事件驱动的支持;增加实时服务质量(QoS)的支持,中间件支持实时 QoS 的表达和执行;增加实时应用编程支持;进行必要的裁剪,以适用于嵌入式系统;进行性能优化。

小结 随着技术的提高和应用发展的需要,实时性成为开放系统中的重要的内含技术之一。从传统的各种实时系统,到新兴的 Internet 和 Intranet 多媒体

(下转第 83 页)

Web 分布式对象计算模型,基于 XML 和 XML 应用的下一代 Web 结构,然后介绍了 WAP 的体系结构和关键技术,最后介绍了 HTTP-NG 的构成和设想。

### 参考文献

- Object Management Group. A Discussion of the Object Management Architecture. June 1997. Available online at <http://www.omg.org/library/omaindx.html>
- Khare R, Tifkin A. XML: A Door to Automated Web Applications. *IEEE Internet Computing*, 1997, 1(4): 78~87
- Goldfarb C F, Prescod P. *The XML Handbook*. Prentice Hall PTR, 1998
- Wales M G. WIDL: Interface Definition for the Web. *IEEE Internet Computing*, 1999, 3(1)
- Gellersen H W, Gaedke M. Object-oriented Web Applications Development. *IEEE Internet Computing*, 1999, 3(1): 60~68
- William C, Janssen JR. A Next Generation Architecture for HTTP. *IEEE Internet Computing*, 1999, 3(1): 69~73
- Wood L. Programming the Web: The W3C DOM Specification. *IEEE Internet Computing*, 1999, 3(1): 48~54
- Manola F. Technologies for a Web Object Model *IEEE Internet Computing*, 1999, 3(1): 38~47
- Manola F. Towards a Web Object Model. [tech. report]. Object Services and Consulting Inc., 1998. Available online at <http://www.objs.com/OSA/wom.htm>
- Manola F. Some Web Object Model Construction Technologies. [tech. report]. Object Services and Consulting Inc., 1998. Available online at <http://www.objs.com/OSA/wom-1.htm>
- Bray T, et al. Extensible Markup Language (XML) 1.0. W3C Recommendation, World Wide Web Consortium, 1998. Available online at <http://www.w3c.org/TR/REC-xml>
- Bray T, et al. Namespaces in XML. W3C Working Draft, World Wide Web Consortium, 1998. Available online at <http://www.w3c.org/TR/WD-xml-names>
- Otte R, Patrick P, Roy M. *Understanding CORBA*. Prentice Hall PTR, 1998
- Simon St. Laurent. *XML: A Primer*, 2<sup>nd</sup> Edition, IDG Books, Foster City, Calif., 2000
- Lassila O, Swick R R, et al. Resource Description Framework (RDF) Model and Syntax. W3C Working Draft, World Wide Web Consortium, 1998. Available online at <http://www.w3c.org/TR/WD-rdf-syntax>
- Berners-Lee T. Metadata Architecture. 1997. Available online at <http://www.w3c.org/DesignIssues/REC-xml>
- Merrick R, Allen C. Web Interface Definition Language (WIDL). W3C Note, World Wide Web Consortium, 1997. Available online at <http://www.w3c.org/TR/NOTE-widl>
- Pitts N. XML: In Record Time. SYBEX Inc., Alameda, CA94501, 1999
- Nielsen H F, et al. HTTP-NG Overview Problem Statement, Requirements, and Solution Outline. IETF Internet Draft, 1998. Available online at <http://www.w3.org/Protocols/HTTP-NG/1998/11/draft-frystyk-httpng-overview-00>. This is work in progress.
- WAP Forum, the WAP 1.2 Specification Suite. WAP Forum, 1999. Available online at <http://www.wap-forum.org/what/technical.htm>
- Hjeltn J, Martin B, King P. WAP Forum-W3C Cooperation White Paper W3C Note. [tech. report]. W3C, 1998. Available online at <http://www.w3.org/TR/NOTE-WAP>

(上接第60页)

服务,都需要实时的支持,当前人们对开放和实时相结合的研究尚处于起步阶段。本文通过对实时系统的特征和开放系统环境的特点的分析,指出一个全局实时的开放系统包含两个方面:通信的实时性和端系统的实时性,并提出通过实时域的划分来对系统进行“分而治之”的方法。文中最后给出了开放实时系统的设计原则和实现模型。利用中间件来实现支持实时的开放系统是当前分布式和实时领域共同关注的问题。CORBA 3.0规范中增加了实时规范<sup>[8]</sup>,这一举措有力地推动了开放系统中实时性的实现。

### 参考文献

- 刘锦德. 对开放系统内涵的澄清. *计算机应用*, 1997, 17(6): 1~4
- Furht B, Halang W A. A Survey of Real-Time Computing Systems. *International Journal of Mini and Microcomputers*, 1994, 16(3): 1~16
- Gill C D, Kuhns F, Schmidt D C. Applying Adaptive Real-Time Middleware to Address Grand Challenges of COTS-based Mission-Critical Real-Time Systems. In: *Proceedings of the 11th IEEE International Workshop on Real-Time Mission-Critical Systems*. Nov. 1999
- ISO/IEC and ITU. Reference Model of ODP. Part 1 Overview. May 1995
- Schmidt D C, Harrison T H, Mungee S. the Design and Performance of Real-Time Object Request Brokers. *Computer Communication*, 1998, 21(4): 294~324
- Mehra A, Indiresan A, Shin K G. Structuring Communication Software for Quality-of-Service Guarantees. *IEEE Trans. On Software Eng.*, 1997, 23(10): 616~634
- Stankovic J A. Misconceptions About Real-Time Computing. *IEEE Computer*, 1988, 21(10): 10~19
- Object Management Group. Realtime CORBA Joint Revised Submission, OMG Document orbos/99-02-12 ed., March 1999