

C/S 系统演化为 B/S 系统的策略和技术^{*}

魏镇韩¹ 赵洪华¹ 陈 鸣²

(解放军理工大学通信工程学院研究生2队 南京210007)¹

(解放军理工大学指挥自动化学院计算机与指挥自动化系 南京210007)²

摘 要 随着网络应用的不断普及,B/S 模式的系统由于其具有的优点被越来越多地采用,已经成为开发应用程序的首选模式。与此同时,大量已经存在的优秀传统 C/S 模式的系统也面临着改造为 B/S 模式的系统的问题。本文研究了由 C/S 模式系统演化为 B/S 模式系统的几种策略,并结合开发一个实例探讨了相关技术问题。

关键词 C/S 模式,B/S 模式,演化策略

Strategies and Technologies of C/S Systems Evolving into B/S Systems

WEI Zhen-Han¹ ZHAO Hong-Hua¹ CHEN Ming²

(Postgraduate Team 2 ICE,PLAUST,Nanjing 210007)¹

(Department of Computer Science and Command Automation ICA,PLAUST,Nanjing 210007)²

Abstract With increasing employment of Web applications,more and more B/S systems are used because of their benefits. B/S model has been the primary model to develop applications. At the same time,many legacy systems using C/S model face the problem of developing from C/S systems to B/S systems. We investigate several evolvement strategies and discuss some related technologies by exploring an instance system.

Keywords C/S model,B/S model,Evolvement strategy

1 概述

Web 平台是一个以客户为中心的应用程序平台,它是一个分布式、开放、适用性强、高性能的平台,随着网络规模的扩大,B/S 模式系统因为客户端零维护的优点已经成为网络应用程序的发展趋势。在另一方面,许多传统的 C/S 模式系统因其维护复杂,升级困难的缺点需要向 Web 方式移植。

C/S 模式系统将应用程序分为客户端和服务端两个部分,其优点是:①客户端和服务端交互能力强。可通过各种通信协议进行交互,交互方式多样且限制较少。②客户端有较强的处理能力,客户端的用户界面可按用户需求定制。其缺点是:①客户端实现和配置较为复杂。②通用性、开放性较差,跨平台支持不足。③维护复杂,升级困难。

B/S 模式将应用程序分为浏览器和服务端两个部分,它是随着 Internet 技术的兴起,对 C/S 体系结构的一种变化或者改进的结构。其优点是:①“零客户端”安装和配置,系统安装、修改和维护全在服务器端解决,用户仅需要一个浏览器就可运行全部的模块。②通用性、开放性较好,对操作系统和平台要求不高。③自动升级方便。④用户可在网络任意节

点接入访问,使用方便。其缺点是:①客户端和服务端交互能力弱,仅使用 HTTP 协议进行数据交互。②功能较弱,难以完全实现传统模式下的特殊功能要求,安全性控制较难。

两种模式互有优缺点,C/S 模式适用于以下情况:较为复杂的数据分析处理,交互性强的应用,复杂的图形图像显示,较高要求的实时性数据传输,局域网环境;而 B/S 模式适用于以下情况:跨平台应用,信息发布和浏览,跨网络应用,客户端零配置。

2 C/S 模式系统变 B/S 模式系统的演化策略

现代应用程序一般基于三层结构模型。逻辑上,三层结构的 C/S 模式系统将应用功能分成表示层、事务逻辑层和数据层三个部分。表示层是应用的用户接口部分,接受用户输入,向事务逻辑层发送服务请求,显示处理结果数据。事务逻辑层执行业务逻辑,向数据库发送读写请求。数据层对应于数据库管理系统,执行数据逻辑,负责管理对数据库数据的读写。C/S 模式和 B/S 模式都可实现三层结构模型,在由 C/S 模式向 B/S 模式移植时首先应考虑它们是实现三层结构时的不同。

^{*})本文工作得到国家“863”计划(No. 2001AA112090)和江苏省自然科学基金(No. BK2001022)的资助。魏镇韩 硕士研究生,研究方向为分布式计算。

C/S 模式的三层结构在物理上的分法可以有多种组合,表示层位于客户机,而数据层位于数据库服务器,事务逻辑层可以位于客户机也可位于应用服务器,或者一部分位于客户机而另一部分位于应用服务器。应用服务器和数据库服务器可以位于一台主机也可位于不同主机。一般客户机和服务器位于不同主机。B/S 模式的三层结构中,表示层和事务逻辑层位于 Web 应用服务器,表示层的一部分也可位于客户端。数据层位于数据库服务器。可以看出,B/S 模式与 C/S 模式在三层结构上的差别在于表示层和事务逻辑层实现的位置不同。B/S 模式中客户端仅实现有限的表示层功能,重要的表示层功能和事务逻辑层功能都在 Web 应用服务器中实现。而 C/S 结构中,客户端有着较强的功能,不但实现表示层的功能,亦可以实现事务逻辑层的功能。

一般而言,C/S 模式系统变为 B/S 模式系统的演化策略有如下几种方式。

2.1 Web Services 封装策略

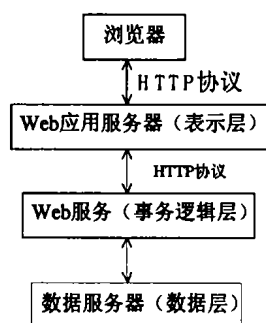


图1 Web Services 封装策略

Web Services 是一种新的计算模型,将紧密耦合的、高效的 n 层计算技术与面向消息的、松散耦合的 Web 概念相结合了起来,可视为 Web 上的可编程组件。Web 服务是为其他应用程序提供数据和服务的应用程序逻辑单元^[1]。应用程序通过诸如 HTTP、XML 和 SOAP 等标准 Web 协议和数据格式来访问 Web 服务。

这种演化策略将 C/S 模式系统的服务器提供的主要功能封装为 Web Services,实现事务逻辑层,可以在 Web 应用服务器动态生成的网页上调用相关 Web Services。Web Services 接受 HTTP 请求,完成相应功能并以 XML 格式的报文将结果返回给调用者。

这种策略适用于以下几种情况:需要跨越防火墙进行通信,因为 Web Services 通过 HTTP 通信,能有效穿过防火墙的限制;基于异构平台的应用,集成各种使用不同语言和平台开发的应用程序,Web Services 提供了一种通用的应用程序集成方法,以 XML 描述数据,以 WSDL 定义接口,以 UDDI 找出可用接口,以 SOAP 调用服务^[2],通过这些标准协

议解决程序之间的通信和集成;需要为不同客户端程序提供统一的调用方式,无论是浏览器还是专用客户端程序甚至移动设备都可以访问 Web Services。

这种策略存在以下缺点:通常需要大量的 CPU、内存和网络资源,因为封装数据的 XML 报文存在一定的数据冗余而且要经过多次处理才能被系统使用,如果系统运行在局域网环境,使用 Web Services 效率较低,可使用 TCP、DCOM 等其他更有效率的方式;Web Services 调用粒度较粗,一次调度返回一个结果,因为 Web Services 使用 HTTP 协议,HTTP 协议是无状态的,每次调用请求都必须建立一个新的 TCP 连接,所以调用粒度较粗。特别是,对于那些不能够直接用 Web Services 封装原有 C/S 结构程序情况,这种演化策略工作量十分大。

2.2 利用代码模块重新开发策略

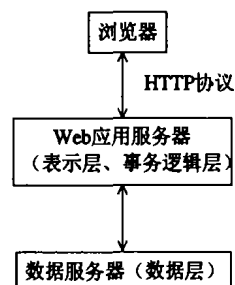


图2 利用代码模块重新开发策略

这种策略重新开发 B/S 系统,仅从代码层次利用原有 C/S 系统的功能,将大部分事务逻辑层功能在 Web 应用服务器中实现。这实际上等于重新开发了一种与 C/S 系统功能等价的 B/S 系统。这种策略适合于以下情况:信息发布系统,因为主要功能是数据库的查询和更改,可以直接使用动态网页的数据库连接技术进行编程;运行于局域网的 C/S 系统,系统功能简单,实现容易。不适合以下情况:功能复杂的大型 C/S 系统,重新开发工作量较大;实时监测系统,Web 应用服务器难以实现复杂的交互通信。

2.3 C/S 和 B/S 模式兼容策略

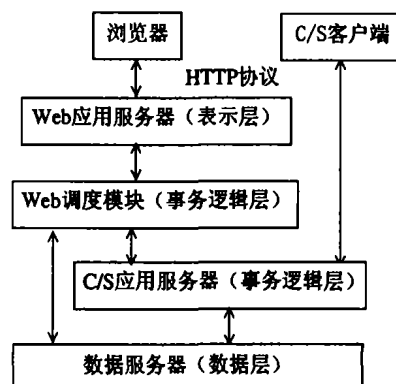


图3 C/S 和 B/S 模式兼容策略

如果现有的 C/S 应用程序功能复杂,并且已经相当成熟和可靠了,在无法直接采用 Web Services 策略的情况下,利用代码模块完全重新开发 B/S 模式系统又因工作量较大而无法实施的话,采用 C/S 和 B/S 模式兼容策略演化将是一个不错的解决方案。

这种策略的主要思路是:

(1)充分利用现有 C/S 系统的大量复杂代码处理资源,保留原有 C/S 系统的服务器作为新系统的共同服务器;

(2)采用 Web Services 技术增加 Web 表示层和 Web 调度模块功能,增加那些能够通过操作数据库完成的功能;通过修改原系统的部分源代码,增加响应 Web 应用服务器服务请求的功能,使之能利用原有代码完成复杂处理功能。

(3)经改造的 C/S 系统的服务器作为两种模式共同的服务器运行,它能为两种模式同时提供服务。

B/S 中的 Web 应用服务器中要实现原来 C/S 客户机的表示层功能,同时向 Web 调度模块发出服务请求。也就是说,B/S 系统中的事务逻辑层分为两个部分:新增加的 Web 调度模块和原有的 C/S 系统服务器,Web 调度模块作为 Web 应用服务器和原有 C/S 系统服务器之间的通信桥梁。

这种实现方式的优点是充分利用原有程序,高效可靠地实现向 B/S 模式演化的工作,并且可以基本实现原系统的所有操作功能。实现上灵活性高,能根据具体情况合理地分配事务逻辑层功能实现的不同位置,比如一些较为简单的交互性功能可尽量放在 Web 应用服务器实现,而复杂的交互功能则要利用原有 C/S 系统服务器的功能。工作量小,使用范围广泛,适合于较为大型的 C/S 系统的移植。这种策略不适合功能较为单一的信息发布系统向 B/S 模式演化,此类系统可使用重新开发策略,直接在 Web 应用服务器实现全部事务逻辑层功能。

3 C/S 和 B/S 模式兼容策略的应用实例

3.1 系统结构

《互联网性能监测系统》是解放军理工大学指挥自动化学院开发的一种端到端、定量、可视化的互联网性能监测系统。该系统能够并行实时地监测大量网络链路的性能参数,如往返时延、带宽、时延抖动、丢包率等并进行数据分析,系统还具有获取分析设备 SNMP 参数的功能。它主要采用 C/S 工作模式工作^[3]。随着该系统更为广泛地应用,用户强烈需要为系统增加 B/S 工作模式。为此,我们需要将该系统演化为 B/S 模式的系统。

B/S 模式下,整个系统分为三层结构:表示层,事务逻辑层和数据层。表示层由浏览器和动态 Web

页面组成。事务逻辑层包括两个部分,一部分是将现有的“互联网性能监测系统”作为后台服务器运行,另一部分是开发一个 Web 调度响应模块,调度后台服务器的相应测量功能,响应 Web 网页的测量请求并返回结果。数据层对应于数据库服务器,数据库服务器实现对数据库的管理和库中测量数据的读写。图4是整个系统的总体框架图。

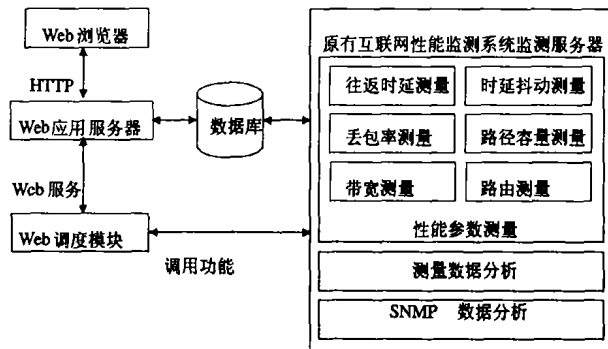


图4 系统的总体框架图

Web 应用服务器实现动态网页的生成,移植原有桌面系统的所有界面,提供 Web 操作界面。Web 应用服务器通过两种方式获得测量数据,一种是直接读取数据库中的数据,另一种是 Web 应用服务器根据用户的测量请求向 Web 调度模块发出 Web 服务请求,由 Web 服务返回结果数据。Web 应用服务器根据数据库中的数据或是 Web 服务提供的数据生成动态页面,将结果返回浏览器。第一种方式应用于历史数据和统计数据的图表显示,以及测量域的 RTT 值和丢包率的显示。第二种方式应用于网络拓扑的发现和路由节点的探测。

Web 调度模块通过调用后台服务器的相应测量功能提供 Web 服务,响应 Web 应用服务器的 Web 服务请求。

C/S 模式的监测服务器作为一个独立的程序运行,经过改造,它能响应测量功能的调用并输出结果。

数据库服务器存储相关测量数据和用户信息,为网络的质量观察和性能分析提供必要信息。

3.2 实现的技术

3.2.1 ActiveX 和页面定时刷新 实时数据的显示通常采用两种方式,一种是通过定时查询的方式获得实时产生的数据,这种方式不需要在客户端下载插件,但是会造成页面的刷新闪烁,不能实现较为复杂的交互功能,适用于短期的实时显示。另一种是通过 ActiveX 或是 Java Applet 技术实现实时显示,这种方式需要在客户端下载 ActiveX 或是 Java Applet,但是实时显示效果好,能实现较为复杂的交互功能。本系统中往返时延、丢包率显示,网络拓扑显示采用 ActiveX,路由探测采用页面定时刷新的方式。

3.2.2 ASP.NET Web 应用服务器采用 ASP.NET 技术构建。ASP.NET 是 Microsoft 公司提供的一种强大的 Web 开发平台,基于服务器端技术来开发动态网页。ASP.NET 技术能实现界面和后台编码的分离,简化了动态网页的编写。

3.2.3 Web 服务 Web 调度模块采用 Web Services 技术构建,采用这种技术的原因是因为它能较好地集成 C/S 系统服务器的一些功能。虽然 Web 服务有许多优点,但在实际应用中却存在着响应时间慢,占用资源多的缺点。在实现中可通过 Web 服务异步调用等方式提高 Web 服务的效率。

总结 C/S 模式和 B/S 模式在网络应用中各

有优缺点,B/S 模式因为使用方便,升级简易成为应用程序的发展趋势。在传统 C/S 模式应用较为成熟的情况下,一些 C/S 模式应用程序面临向 Web 移植的问题,本文讨论了由 C/S 模式系统演化为 B/S 模式系统的策略,并结合开发一个网络监测系统的实例探讨了相关技术问题。

参考文献

- 1 Cerami E. Web Services Essentials[M],O'Reilly,2002
- 2 Chappell D. 侯杰,荣耀译. NET 大局观-Understanding . NET [M]. 华中科技大学出版社,2003
- 3 陈鸣,等. 分布式网络性能监测系统. 电信科学,2003(5):60~63

(上接第64页)

新注册仓中服务的动态信息。以第二种方式为例来说明具体的订阅操作(如图3),订阅方 VO-IS 通过

实现 NotificationSink 接口获取通知的消息,被订阅方通过实现 NotificationSource 接口对消息的订阅行为进行管理。

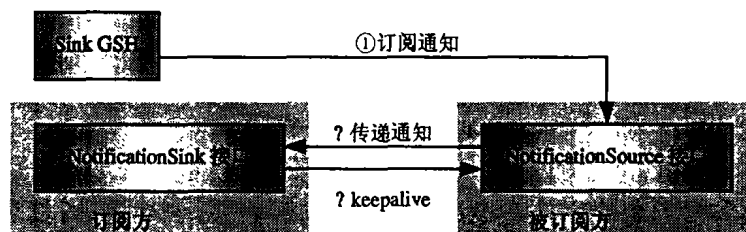


图3 订阅网络服务的过程

4 Dm-IS

由于 VO-IS 是针对某一特定联盟、组织建立的,可以动态建立和撤销,稳定性较差,存储能力也有限。因此,系统引入 Dm-IS 对整个网格服务空间进行协调管理,它的存在是稳定的,是元信息管理机制中的最高层服务。在结构上它增添了全局域节点信息库,用来记录全局域节点的位置等相关信息。

Dm-IS 的元信息管理策略与 VO-IS 基本相似,对于相同点文中将不再赘述,下面着重介绍它们间的不同之处:第一,Dm-IS 与 VO-IS 所存储的服务信息不同,前者是针对虚拟组织的服务,而后者是针对资源服务;第二,Dm-IS 与 VO-IS 接收的查询请求方或订阅请求方不同,前者是 VO-IS,而后者是资源层的服务;第三,当本地 Dm-IS 不能满足查询请求时,语义分析器依据全局域节点信息利用通知机制将查询请求转发到其它域继续进行查询,而 VO-IS 是直接将查询请求转发到所属的 Dm-IS;第四,它们实现的目的不同,Dm-IS 是查找能继续执行查询请求的 VO-IS,而 VO-IS 是查找最终满足用户需求的资源层服务。

结束语 元信息管理技术是网格系统的核心之一,是实现网格其它功能的重要基础。本文建立一个基于层次的支持网格服务的元信息管理机制,完成

对网格环境中信息的发布、发现、监控与更新。在此基础上提出智能化的查询策略,选择功能最匹配、性能最优的服务作为查询结果,极大程度地提高了查询的全面性与精确性。此元信息管理机制已经应用于东北大学软件研究所承担的863项目——ICES-Grid (Intelligent Composition for E-business Service Grid)中,经验证,该机制具有良好的安全性、精确性与可扩充性。下一步我们将对服务匹配算法与服务间的通信机制做进一步完善。

参考文献

- 1 Kesselman F C, Nick J, Tuecke S. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. Globus Project. <http://www.globus.org/research/papers/ogsa.pdf>, 2002
- 2 ARIBA INC. UDDI Technical White Paper. <http://www.uddi.org>, Sep. 2000
- 3 Mario Cannataro. Knowledge Discovery and Ontology-based services on the Grid. University 'Magna Græcia' of Catanzaro, 88100 Catanzaro, Italy
- 4 Wu Zhaohui, Chen Huajun, Xu Jiefeng. Knowledge Base Grid: A Generic Grid Architecture For Semantic Web. May, 2003
- 5 GT3 Index Service User's Guide. <http://www.globus.org/ogsa/releases/final/docs/infosvcs/indexsvc-ug.html>
- 6 Silva V. Querying the Grid with the Globus Toolkit monitoring and discovery service. Software engineer, Internet technology, IBM, April, 2003