

分层架构体系研究*

丁 勇

(云南爱因森软件职业学院 昆明 651701)

摘 要 分布式多层 B/S 架构克服了 C/S 结构和局域网为中心、系统难以扩展、维护成本高、软件移植性差等缺点,它把表示逻辑、业务逻辑、数据库等操作分离开来,各层相对独立。这种多层分布式计算模型适应了企业应用的发展要求,使企业计算具有良好的可扩展性、可靠性、稳定性、安全性等特点。同时,通过组件(Component)技术的使用,缩短了系统开发的周期和降低了开发成本,大大降低了企业应用系统开发的复杂度。

关键词 三层架构 插件 分布式架构

中图分类号 TP311.5

文献标识码 A

文章编号 100531-5411

Research on the Hierarchical Architecture System

Ding Yong

(Yunnan Einsun Software College Kunming 651701)

Abstract The distributed multi-tier B/S structure overcome the shortcomings of C/S structure, as the center of LAN, the system is hardly to expand, high maintenance cost, and poor portability of software etc. which separated from the operations of presentation logic, business logic and database etc., each hierarchy is independent respectively. This multi-tier distributed computing model adapted the requirements of applied development in enterprises, so as to compute in enterprises possessed the features of good expandability, reliability, stability and security etc. Meanwhile, through the use of Component technology, it's shortened the system development cycle and reduced the development costs as well as decreased greatly to the complexity of applied system development in enterprises.

Keywords 3-tier architecture Plug-ins Distributed architecture

3/N 层架构是经典的多层架构模型,对于稍微复杂一点或特别复杂的系统,不使用分层架构是很难想象的。图 1 是经典的 3 层架构,这确实是解决系统复杂性的一种主流模式,但是,只要采用了 3/N 层架构是不是就一定能解决系统的复杂性,关键在于你在你的系统中如何实作你的 3/N 层结构。



图 1 3/N 层架构

在采用了 3/N 层架构后,我们还是要解决以下非常重要的问题:系统的可扩展性(能从容地应对变化)、系统的可维护性(因为系统并不是使用一次就被抛弃)、方便部署(在需求变化时,方便部署新的业务功能)、还有等等其它系统质量属性。然而系统的可扩展性和可维护性是大多数软件系统必须解决的重中之重,这是由于当前需求复杂多变的软件环境决定的。

一、“框架+插件”架构

经典的 3/N 层架构是对系统进行“纵向”分层,而“框架+插件”架构对系统进行“横向”分解。3/N 层架构和“框架+插件”架构处于一个平等的位置,它们没有任何依赖关系。但是我经常将它们结合在一起使用,我们的系统在经过 3/N 层架构的纵向分层和“框架+插件”架构的横向分层后,可以被看作一个“网格”结构,其中的某些网格可以看作是“扩展点”,我们可以在这些扩展点处挂接“插件”。也就是说我们可以在 3/N 层架构的每一层都挂接适当的插件来完成该层的一些功能。如图 2 所示:

插件最主要的特点是可以实现“热插拔”,也就是说可以在不停止服务的情况下,动态加载/移除/更新插件。所以,采用插件技术可以实现以下功能:

在 UI 层,我们可以在运行时,替换掉某些用户界面、或加载与新的业务相关的用户界面。在业务逻辑层,我们可以在运行时

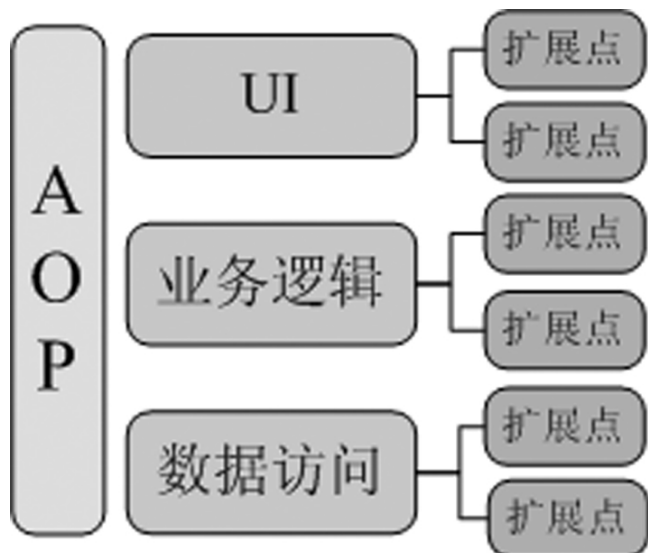


图 2 框架 + 插件架构

加载、替换或者删除某项业务服务。在数据访问层,通过使用插件技术我们可以动态地添加对新的数据库类型(如 Oracle、MySQL)的支持。插件的“热插拔”功能使得我们的系统有非常好的可扩展性。

如果我们需要升级系统,很多情况下,只要升级我们的插件(比如业务插件)就可以了,我们可以做到在服务运行的时候进行插件的自动升级。

要想将系统做成“框架+插件”的结构,要求我们需要在系统的各层进行“松耦合”设计,只有松耦合的组件才可以被做成“插件”。

在 3/N 层架构中融合“框架+插件”架构,最难的是对业务逻辑层的松耦合处理,这需要我们细致分析业务需求之间的关联,将耦合度紧密的业务封装在一个组件中,如此得到的相互独立的业务组件便可以有机会成为插件。这个过程可能需要不断的重构、设计重构。

相比于那些紧密耦合的组件,松耦合的组件更加清晰明确、更加容易维护。另外,在该架构模型中引入了 AOP 框架进行 Aspect 焦点的集中编程(比如处理日志记录、权限管理等方面),使得 Aspect 代码不会掺杂在正常的业务逻辑代码中,使得代码的清晰性、可维护性进一步增强。

从上述介绍可以看出,采用 3/N 层架构和“框架+插件”架构相结合,我们可以增强系统的可扩展性、可维护性和简单部署升级的能力。

二、地域分布式架构

地域分布式架构主要针对类似基于位置的服务(LBS)的需要进行地域分布的应用。地域分布式架构基于上述的 3/N 层架

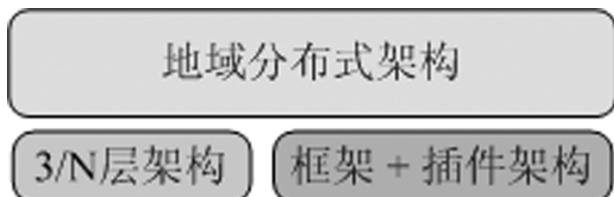


图 3 基本地域分布式架构

构和“框架+插件”架构,它们的关系如图 3 所示:

假设我们需要为全国的各个大城市提供我们的业务功能服务,假设每个城市的客户量很大,而且每个城市访问的数据可能是不一样的(如每个城市的手机、电话数据)、访问的功能也不尽相同(如有的城市提供考勤查询服务,而另一些城市不提供)。客户除了跟我们的系统请求服务之外,可能还想通过我们的系统与他的用户进行即时通信,而它们用户可能与他在同一个城市,也可能位于另外一个城市。

首先,地域分布式架构将用户管理和业务功能服务分开,分别由应用服务器(AS)和功能服务器(FS)负责,然后将它们部署到不同的节点上。AS 和 FS 都采用了 3/N 层架构和“框架+插件”架构相结合的架构,比如,FS 通过功能插件提供功能服务,如图 4 所示。

对于昆明这个地域,我们部署了一台 AS 和一台 FS,客户端

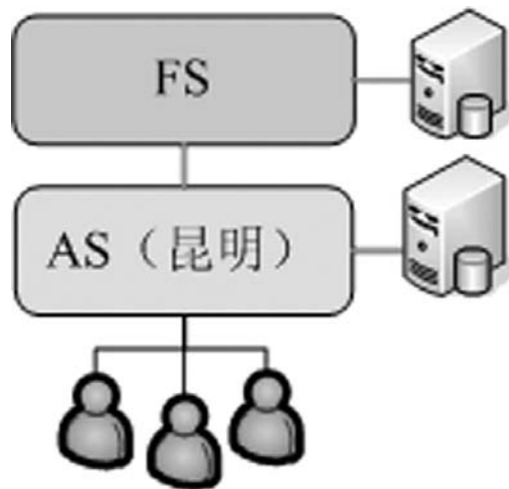


图 4 框架+插件的地域分布式架构

通过连接到 AS 进行服务请求。假设有一天,我们在昆明的客户急剧增加,这时压力最大的是 FS,因为所有的业务计算都是在 FS 上完成的。地域分布式架构将允许我们在不停止任何服务的情况下,动态的添加 FS 服务器,新添加的 FS 服务器会自动注册到 AS,如图 5 所示。

AS 可以监控每个 FS 的负载(如 CPU 消耗、内存消耗),再

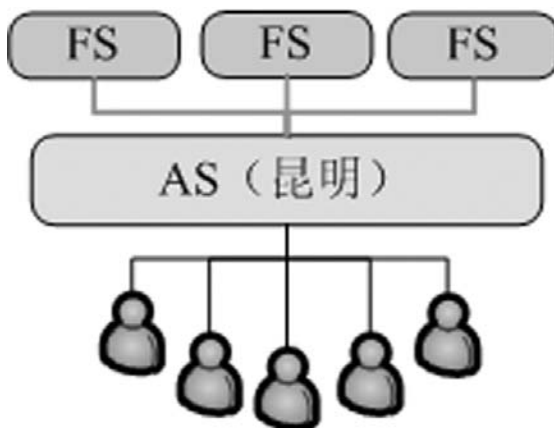


图 5 多个功能服务器的地域分布式架构

(下转第 62 页)

用户间具有更强的交互性。特别强调的是,系统实现了一套完整的 Ajax 查询机制。在服务器端,查询结果被封装成 Java 类,再由 Java 类生成 XML 格式的字符串返还给客户端。在客户端,实现了 XML 文件的解析,将结果解析成 JavaScript 对象,这样 JavaScript 对结果的处理更加方便,整个机制也很灵活。

在实际的网站开发中,程序员仅仅负责的是后台实现,前台页面组织和样式需要美工完成,但是在本项目中前台页面和后台实现均由笔者完成,因此前台页面组织和样式可以由专业人士进行进一步的提高和改善。

参考文献

- [1] <http://baike.baidu.com/view/733.html?wtp=tt> [EB/OL], 2009-5-3.
- [2] 王珠珠, 张伟远. 网上教学平台及网站建设现状分析 [J/OL]. 中国教育和科研计算机网, 2005; [2009-04-18]. http://www.edu.cn/li_lun_yj_1652/20060323/t20060323_119802.shtml.
- [3] 戎伟, 张双. Struts- Java 流行服务器、框架、工具及整合应用 [M]. 北京: 人民邮电出版社, 2006: 3~11.
- [4] 孟劼. Spring- Java 轻量级架构开发实践 [M]. 北京: 人民邮电出版社, 2006: 1-6.

(上接第26页)

有客户端请求到来时, AS 会将请求交给负载最低的 FS 处理, 这就实现了 FS 的负载均衡。如果 Client A 需要与 Client B 进行即时通信, 那么这些通信消息将通过 AS 中转。上面看到的是我们的系统在昆明的部署, 而在其他城市部署情况也一样, 如图 6 所示。

在这种情况下, AS 和 AS 之间是相互独立的, 但是经常会发生 AS 之间需要相互通信的情况, 比如: Client A 需要与 Client B 进行即时通信, 或者 Client A 需要请求上海地区独有的服务

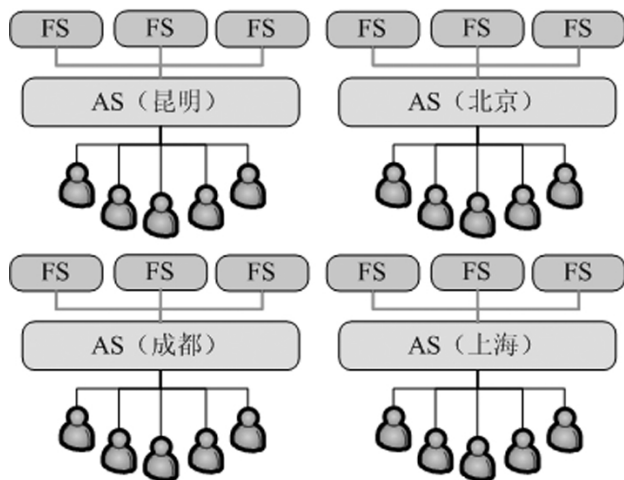


图6 地域分布式架构

等等。如图 7 所示, 地域分布式架构使用跨区域的应用服务器 (IRAS) 来解决 AS 之间的通信问题。所有 AS 在启动的时候, 将自动向 IRAS 注册。

如果, 我们想在重庆也提供我们的服务, 那么我们只需要在重庆部署我们的 AS 和 FS, 这样就可以融入到图 2-10(?) 所示的整个地域分布式架构中。

[5] 李刚, 郭莹, 薛大龙, 等. 贯通 Hibernate 开发 [M]. 北京: 电子工业出版社, 2008: 2~27.

[6] 梁立新. 基于 Struts- Spring- Hibernate 的 Java 应用开发 [M]. 北京: 电子工业出版社, 2008: 1~50.

[7] Paul R. Reed, Jr. Developing Applications with Java and UML [M]. 北京: 清华大学出版社, 2003: 117~144.

[8] Craig Larman. Applying UML and Patterns [M]. 北京: 机械工业出版社, 2006: 45~75.

[9] Craig Larman. Applying UML and Patterns [M]. 北京: 机械工业出版社, 2006: 100~124.

[10] Jeffrey D. Ullman, Jennifer Widom. A First Course in Database Systems [M]. 北京: 电子工业出版社, 2006: 23~80.

[11] 郭伟伟, 刘端阳. JavaScript 全程指南 [M]. 北京: 电子工业出版社, 2008.

[12] 王沛, 冯曼菲. 征服 Ajax Web 2.0 开发技术详解 [M]. 北京: 人民邮电出版社, 2006.

作者简介

张军, 女, 1960年11月出生, 北京市人, 研究生文化程度, 武警北京指挥学院训练部实验中心, 高级工程师。

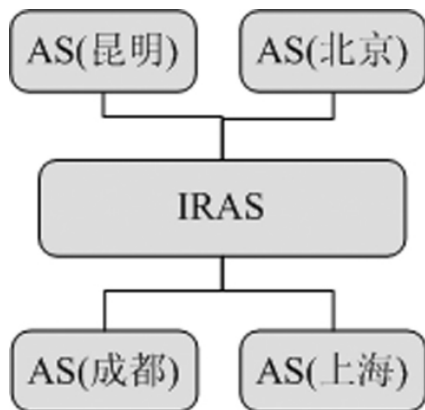


图7 IRAS 架构

参考文献

- [1] 朱三元, 钱乐秋, 宿为民编. 软件工程技术概论 [M]. 科学出版社, 2002.1.
- [2] 杨一平等著. 现代软件工程技术 with CMM 的融合 [M]. 人民邮电出版社, 2002.11.
- [3] 邵维忠, 杨芙清编著. 面向对象的系统分析 [M]. 清华大学出版社, 广西科技大学出版社.
- [4] 梁爱虎编著. 基于服务总线的 Struts+EJB+Web Service 整合 [M]. 电子工业出版社, 2007.1.

作者简介

丁勇, 男, 1975年生, 2007年获云南大学软件工程硕士, 讲师职称。参加工作后主持或开发多个大型企事业单位的电子政务或企业信息化项目, 典型项目有企业门户系统、办公自动化系统、短信平台、客户关系管理系统、酒店管理系统等。2006年开始授课, 主要讲授 ASP.NET、数据库、项目开发、软件工程等软件课程。工作十年有余, 具有软件设计、架构、项目管理经验, 主要研究方向为软件工程。