广西科学院学报
Journal of Guangxi Academy of Sciences

2010,26(4):479～482
Vol. 26, No. 4 November 2010

一种可复用的网络社交游戏开发框架设计
Design for Reusable Application Framework of Social Game

顾　林[1],唐昭琳[2]
GU Lin[1], TANG Zhao-lin[2]

(1.广西大学计算机与电子信息学院,广西南宁　530004;2.广西师范学院,广西南宁　530023)
(1. School of Computer, Electronics and Information, Guangxi University, Nanning, Guangxi, 530004, China; 2. Guangxi Teachers Education University, Nanning, Guangxi, 530023, China)

摘要:采用应用开发框架技术,提出一种可复用的网络社交游戏应用框架,避免底层关键技术的重复开发。利用该框架开发1个网络社交游戏游戏并上线运营,实践表明该框架的架构的可伸缩性、可用性较好,可以缩短开发周期。

关键词:网络社交游戏　应用框架　开发
中图法分类号:TP391　　**文献标识码**:A　　**文章编号**:1002-7378(2010)04-0479-04

Abstract: The solution of scalable network architecture to improve workload ability for social games is proposed based on web clusters, load balancing on web layer and asynchronous duplication on data layer. A reusable social game application framework is also proposed. A social game was developed and on-line operated based this framework. The experimental results show this approach is efficiency.

Key words: social game, application framework, development

互联网的普及带动了基于社会关系网络的社交网站(Social Network Site,SNS)的发展。网络社交游戏(Social Game)是网页游戏与社交网站结合的产物,它依托大型的SNS网络社区,游戏操作简单,强调好友之间的互动,成为游戏市场中增长最为迅速的一个门类,具有很大的市场活力和潜力。

与传统大型网络游戏不同,网络社交游戏基于Web技术,依赖于SNS提供的开放平台接口(OpenAPI),属于基于浏览器内核的微客户端游戏。网络社交游戏要求用户体验好而且客户端下载量小、占用系统资源少,服务端处理的逻辑量大,支持多社交平台、大用户量、高并发访问等。目前,网络社交游戏前端普遍采用富互联网应用技术(Rich Internet Application,RIA)。基于RIA技术、可扩展性、可复用性、用户体验好的网络社交游戏开发框架是该领域未来研究发展重点。由于网络社交游戏历史不长,目前相关研究成果仍比较少。JGnet[1]是一款基于RIA的商业引擎,包含了客户端和服务器端,全部采用Java撰写,具有跨平台的特点,解决了Java客户端在处理复杂的即时交互需求和图形处理需求时的效率问题。文献[2]介绍的"快乐神仙"游戏引擎由客户端引擎、网络引擎、配置和编辑器三部分组成。客户端采用Flash作为图形播放器,服务器端完成大部分游戏逻辑。客户端采用分层结构,即显示层、逻辑层、数据交互层。逻辑层是一个观察者模式,接受数据通知,对应产生相应的图形显示。服务器端包括连接服务器,游戏服务器、全局消息服务器、数据库服务器。

游戏应用开发框架可看作游戏引擎的重要组成部分。在网络社交游戏的开发中,良好的组件化的、

收稿日期:2010-9-20
收稿日期:2010-10-09
作者简介:顾　林(1965-),女,研究员,主要从事计算机应用研究。

扩展性好的开发框架，能够提高关键技术的复用度，促进游戏软件的快速开发，有助于创建稳定的游戏程序，减少游戏开发者重复编写代码的劳动，从而极大地降低开发门槛，缩短游戏开发周期，提高游戏产出量。本文采用应用开发框架技术，提出一种可以复用的网络社交游戏开发框架。

1 系统架构设计

在大量玩家同时在线的情况下，游戏服务器通常需要面对两个问题：大量的连接以及大量的数据。大量的连接意味着高并发，大量的数据则意味着大量的内存开销、IO 操作及网络流量。当同时在线用户达到一定数量时，传统的服务器优化技术，如线程技术、内存技术，已经基本失效，想要提高负载能力，必须在构架方面着手。另外，游戏系统的可用性，也必需从架构设计方面加以考虑。系统的架构要求是可伸缩的、高可用性的。一个系统的可伸缩性主要通过两种方式实现：一个是横向扩展(Scale out)，另一个则是纵向扩展(Scale up)[3]。横向扩展是通过增加处理节点的方式来提高整体处理能力。纵向扩展是通过增加当前处理节点的处理能力来提高整体的处理能力，如升级现有服务器的配置，增加内存，增加 CPU，增加存储系统的硬件配置，或者直接替换处理能力更强的服务器和更高的存储系统。从长远影响来看，横向扩展有更大的优势，而且是系统达到规模之后的必然趋势。因为单台机器的处理能力总会受到硬件技术的限制，而硬件技术的发展速度是有限的，很多时候很难跟得上业务发展的速度。而且越是高处理能力的高端设计，其性价比总是越差。所以，通过多台廉价的 PC 服务器构建高处理的分布式集群，有节约成本、提高整体处理能力的目标，是一种较为经济可行的解决方案。

根据上述分析，系统物理架构模型可以设计成图 1 所示。游戏系统前端由两台服务器负责负载均衡处理，将访问压力均衡分摊到后端的 Web 服务器。其中一台为主机，一台为备机。主备机之间的监控信息通过“心跳线路”传递。当主机出现问题时，备机能自动接管主机的工作，为系统整体可用性提供保障。游戏系统的 Web 服务器采用集群技术，由负载均衡将访问压力分摊到不同服务节点。当用户量上升至单台压力超负载时，可以随时增加新的服务节点，为系统在 Web 层提的扩可展性、可用性提供保障。游戏系统的数据库服务器采用主从结构模式[4]，一台作为主数据库服务器，若干台作为从数据库服务器。主从数据库服务器通过自动复制保持数据一致。所有写操作对主数据库操行，所有读操作对从数据库操作，进行读写分离，以缓解主数据库压力。当数据异常丢失或软/硬件故障时，能够确保数据不会丢失，保证游戏数据的可用性。

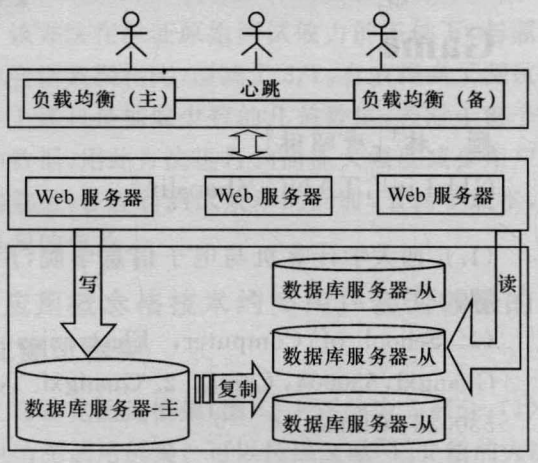

图 1 系统架构模型

2 应用框架设计

框架是一组相互合作的类的集合，规定应用的体系结构，阐明整个系统的设计，各个部件之间的依赖关系、职责分配、相互协作和控制流程，表现为一组抽象类以及实例之间协作的方法，为复用提供了上下文(Context)关系。应用框架设计的好与差直接影响框架的可重用性和可扩展性。框架设计应采取支持客户化的设计思路，使不同的社交游戏应用开发能够使用框架的通用功能，同时，开发者通过在框架中插入自定义的游戏业务逻辑，可以按游戏业务需求剪裁出各自的应用。框架本身如果有很高的可扩展性，就能适应更多类型的游戏应用开发[5]。应用可以降低系统模块间的耦合度[6]的，分解复杂系统最常用的分层技术，网络社交游戏系统的应用框架由位于客户端的客户层和位于服务器端的服务层和数据层组成(图 2)。

客户层负责获取玩家的输入，通过通信协议调用服务端接口层的接口，由接口调用业务逻辑层相应的业务服务组件进行处理，并将业务服务组件处理结果返回给客户层，客户层形成界面在客户端展示。游戏的主要业务逻辑，由业务逻辑层实现，包括获取和更新玩家数据、获取好友信息等。客户层运行在客户端浏览器上，以事件处理为中心，由 RIA 的客户端引擎负责处理执行。主要由主程序模块、资源库和一系列基础组件组成。主程序模块是游戏前端程序的核心控制模块，主要负责游戏资源素材

加载、基础数据配置文件加载、游戏地图加载、玩家数据加载、NPC 加载、场景物品实例创建、场景布局、添加各种事件监听器(即事件处理方法),还包括各种视图类,数据表示类的定义等。主程序是面向事件的,它监听系统的事件,并做出控制的反应。这里的事件主要是用户操作鼠标键盘的事件和定时器事件。它的启动过程被定义为一个模板形式,主要过程为:创建应用程序实例、创建预加载完成监听器、预加载游戏资源及基础数据、加载玩家数据、创建场景管理器、创建场景并初始化部局、创建事件监听器并注册、启动事件循环。游戏资源库由若干个资源列表文件和若干个游戏资源文件包组成。资源列表描述各资源包信息,游戏资源包提供了游戏调用的各种图片、动画、音频资源,比如各种地图、面板、对话框、角色、物品等,以及记录游戏基础数据配置文件。每个资源文件包都是各自独立的,并通过资源列表文件进行索引。基础组件库包括资源加载组件、NPC 管理组件、场景管理组件、物品管理组件、事件管理组件、服务请求组件等,由应用开发框架的核心组件构成。这些核心组件工作在客户浏览器端,主要负责响应玩家的操作、与服务器交互、管理场景中的物品和 NPC、向服务器请求资源等等。基础组件是系统可拆卸的部分,因此可对系统进行增量式开发。

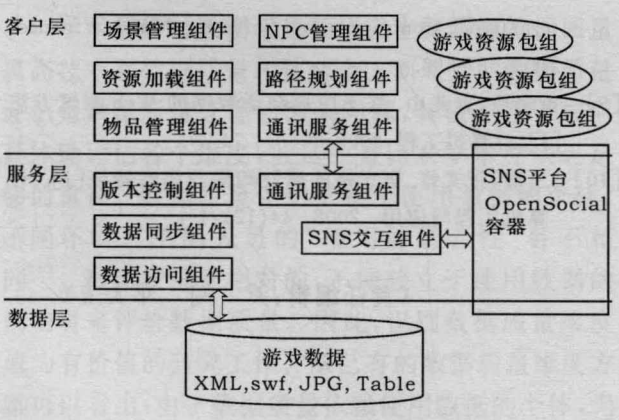

图 2 游戏应用开发框架结构

服务层位于服务器,由业务逻辑接口层、业务逻辑实现层和数据访问层组成。游戏客户端通过接口层,调用服务端逻辑。业务逻辑层主要包括 SNS 交互组件、版本控制组件、数据同步组件、数据访问组件和通信服务组件等。用接口层来为游戏客户端提供统一而清晰的访问服务,传统意义上的表示层由 RIA 应用代替。

3 应用效果

为了验证网络社交游戏开发框架的有效性,在《面馆》游戏开发中采用了框架式开发。《面馆》是一款模拟经营类的娱乐休闲社交游戏,它具备了一般小型社交游戏的特点。游戏包括面馆大厅和厨房两个场景,是接近 45°的视角图。游戏共有 NPC40 多个,NPC 可以场景内行走和对话,可以从门外进来点面、坐下来吃面、付款等。采用开发框架整个开发周期约为 3 个月。系统软件环境:服务器操作系统均采用 64 位 CentOS,数据库系统采用 MySQL 5.1 版本,Web 应用服务器采用 Tomcat 6.0.20,负载均衡采用 Nginx 0.8.28。OpenSocial 采用 0.81 版本。系统硬件环境:Web 服务器(2 台),数据库服务器 2 台,1 台为主数据库,1 台为从数据库,负载均衡服务器 1 台。配置均双 CPU,4 核,16GB 内存。

服务器性能状态主要表现在 CPU、IO 及网络,通过系统 Load 值、CPU 使用率、磁盘 IO 量、网络流量这 4 个监控量来监控。在日活跃用户量为 50000 时,其中 1 台 Web 服务器 1 天的 CPU Load 监控情况(图 3)显示,Web 服务器的负载量不大。当日活跃用户量增长到 200000 时,Web 服务器增加到 4 台,系统 Load 值保持在可接受范围内。

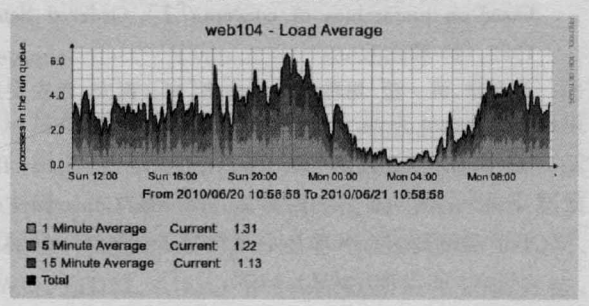

图 3 《面馆》游戏开发中的一台 Web 服务器 CPU 监控情况

4 结束语

随着互联网的普及和 RIA 技术的发展,网络社交游戏将会得到更为广泛应用,迫切需要相关技术的发展和支撑。因此,更好的用户体验、更高性价比的解决方案仍将是该领域的研究重点。为应对大用户量下的高并发访问,在架构的可伸缩性和可用性方面,需要对分布式架构技术做更深入的探讨和实践。本文提出的解决方案是直接采用关系数据库系统,进行读写分离式集群。实际应用表明,当负载突然变大,数据库服务器 IO 性能不足时,该方案的数据库系统会出现数据库复制延迟、主从数据不一致

问题。随着内存型数据库系统[7]、缓存系统以及分布式并行计算技术的发展，游戏应用开发框架的可伸缩性和性能，将会得到更大限度的提升。

参考文献：

[1] 汉森. JGnet技术 [EB/OL]. http://www.handseeing.com, 2009-11-01.

[2] 周勇. 蓝港在线网页游戏引擎[J]. 程序员，2009(9)：60-63.

[3] Cal Henderson. 构建可扩展的web站点[M]. 北京：电子工业出版社，2008.

[4] 简朝阳. MySQL性能调优与架构设计[M]. 北京：电子工业出版社，2010.

[5] Xin Chen. 应用框架的设计与实现[M]. 北京：电子工业出版社，2005.

[6] Martin Fowler. Patterns of Enterprise Application Architecture[M]. Addison-Wesley，2005.

[7] 何坤. 基于内存数据库的分布式数据库架构[J]. 程序员，2010(7)：116-117.

（责任编辑：邓大玉）

（上接第478页）

化，进一步提高测试效率。下一步我们将运用增量式算法对精简集进行动态更新。

参考文献：

[1] Elbaum S, Rothermel G. Leveraging user-session data to support web application testing[J]. IEEE Transactions on Software Engineering, 2005, 31(3): 187-202.

[2] Wille R. Restructuring lattice theory: an approach based on hierarchies of concepts[J]. Ordered Sets, 1982: 445-470.

[3] 张文修，姚一豫. 粗糙集与概念格[M]. 西安：西安交通大学出版社，2006.

[4] Martin P, Eklund P W. Knowledge retrieval and the world wide web[J]. IEEE International Conference on Software Engineering Research, Management and Applications, 2000, 3(15): 18-25.

[5] Paolo Tonella. Using a concept lattice of decomposition slices for program understanding and impact analysis[J]. IEEE Transaction on software engineering, 2003, 29(6): 495-509.

[6] Ping Ng. A concept lattice approach for requirements validation with UML state machine model[C]. IEEE International Conference on Software Engineering Research, Management and Applications, 2007: 393-400.

[7] 李心科，张磊磊. 基于概念分析的用户会话约减技术[J]. 计算机工程，2009, 35(7): 60-63.

[8] Sara Sprenkle, Sreedevi Sampath. An empirical comparison of test suite reduction techniques for user-session-based testing of web applications[C]. IEEE International Conference on Software Mainte- nance, 2005: 587-596.

[9] 黄晓玲，袁兆山. 基于用户会话数据的Web测试方法[J]. 计算机工程，2009, 35(3): 74-79.

[10] 王霞，张文修. 概念格的属性约简与属性特征[J]. 计算机工程与应用，2008, 44(12): 1-4.

（责任编辑：尹 闯 邓大玉）