

Dubbo 服务框架技术在学习系统开发中的应用与实践^①

李磊, 李娟

(北京工业大学 信息学部, 北京 100124)

(北京市物联网软件与系统工程技术研究中心, 北京 100124)

摘要: 随着软件技术和网络的飞速发展, 互联网技术越来越多地被应用到了教育领域, 互联网教育以其日益丰富的形式和内容被越来越多的人接受和采纳。在线学习已经成为一种新型而且有效的学习方式。在线学习以其独有的时空无关性、及时性、可重复性等特点, 使得每一个互联网用户都可以进行在线学习。同时也对在线学习系统提出了一项挑战——如何使学习系统在高并发访问下稳定运行? 本文主要介绍以 Dubbo 分布式架构为后台服务的学习系统的设计与实现。该系统通过教育游戏吸引儿童注意力, 提高学习积极性, 引导儿童通过网络进行有效的学习; 通过 Dubbo 分布式集群架构, 缓解高并发访问时服务器的压力。

关键词: 在线学习; 教育游戏; 高并发访问; Dubbo; 分布式架构

Application of Dubbo Service Framework Technology in the Development of Learning System

LI Lei, LI Juan

(Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China)

(Beijing Engineering Research Center for IoT Software and Systems, Beijing 100124, China)

Abstract: With the rapid development of software technology and network, internet technology has been increasingly applied to the field of education, and internet education has been more and more accepted and adopted by more and more people for its increasingly rich form and content. Online learning has become a new and effective way of learning. With its unique characteristics of the time independent, timeliness, repeatability, online learning can makes every internet user learn online. At the same time, it also presents a challenge to the online learning system that how to make the learning system run stably under high concurrent access. This paper mainly introduces the design and implementation of the learning system based on Dubbo distributed architecture. The system can attract the attention of children through the educational games, improve their learning enthusiasm, and then guide children to learn effectively by network. It can ease the pressure of high concurrent access to the server, through the Dubbo distributed cluster architecture.

Key words: online learning; educational games; high concurrent access; Dubbo; distributed architecture

网络技术及其构成的文化图景已经成为当代教育不可或缺的部分, 高等教育借助在线发展已成不可逆转的趋势^[1], MOOC^[2]就是一个成功的典型。

“教育游戏”一词最早起源于 21 世纪 80 年代的美国, 教育游戏是指在游戏化学习理论和娱教技术的指导下, 以促进儿童学习, 提高儿童学习兴趣为最终目的, 将游戏和知识点相结合而产生的计算机软件系统^[3]。目前, 已有一批教育游戏的实践网站, 例如

网娃(Webkinz)^[4]、企鹅俱乐部(Club Penguin)^[5]、摩尔庄园^[6]等, 都有一定量的儿童用户群。

然而, 目前大多教育游戏只是把游戏和知识点生硬的结合在一起, 内容单调乏味, 缺乏对儿童的吸引力; 并且, 部分家长仍然认为游戏弊大于利, 阻止孩子使用教育游戏网站。这些都是导致教育游戏系统达不到预期效果的原因。本文针对这些问题, 提出了一种教师、儿童、家长三端互联的在线儿童英语学习模

^① 基金项目:北京工业大学资助项目(ER2015C010605)

收稿时间:2016-09-28;收到修改稿时间:2016-11-10 [doi: 10.15888/j.cnki.csa.005810]

式. 教师、儿童和家长三端互联, 既方便了教师教学备课, 也可以通过游戏化的方式吸引儿童学习, 同时还可以帮助家长掌握孩子学习情况, 消除顾虑. 本文提出的在线学习系统主要由 4 个部分组成——教师端, 学生端, 家长端以及后台服务端. 教师端, 学生端, 家长端为应用端, 主要实现功能性应用. 后台服务端为数据服务器端, 采用了分布式服务架构, 使用 Dubbo 与 SpringMVC 的整合架构, 达到负载均衡以及高容错的目的, 并使前端应用能更快速和稳定的响应.

1 系统开发中的关键技术

1.1 Dubbo 分布式架构

Dubbo 是一个分布式服务框架, 致力于提供高性能和透明化的 RPC 远程服务调用方案, 是 SOA 服务化治理方案的核心框架^[7]. 其核心部分包含: 远程通讯, 集群容错和自动发现.

同时, Dubbo 还可以与 Spring 框架无缝集成, 即 Dubbo 采用全 Spring 配置方式, 透明化接入应用, 对应用没有任何 API 侵入, 只需用 Spring 加载 Dubbo 的配置即可, Dubbo 基于 Spring 的 Schema 扩展进行加载.

1.2 Spring MVC 框架

Spring 是一个开源且基于 POJO (Pure Old Java Object) 的轻量级 J2EE 应用框架, 它是为解决企业应用程序开发的复杂性而创建的.

SpringMVC 由以下三个部分组成: 模型(Model)、视图(View)和控制器(Controller).

Model 代表应用的业务逻辑, 包含应用程序的核心功能. View 是应用的视图层, 提供模型的表现. Controller 提供应用的处理过程控制, 并对用户的输入做出反应.

2 系统分析设计

2.1 系统应用端功能设计

本系统主要面向三类用户: 教师、儿童和家长. 针对这三类不同的用户, 设计了 3 个子系统: 教师端、学生端和家长端. 在使用过程中, 教师端、学生端和家长端三端分立, 同时又互相联通, 为教育、学习和监督提供了一条信息流动的回路. 本系统将教育游戏与课堂教学结合起来. 教师在课堂上讲解的知识点, 可以通过教师端系统, 以作业的形式布置给学生; 学生在学生段领取到作业任务后, 又可以通过玩游戏

巩固所学的知识点. 这样, 不但通过游戏提升儿童学习兴趣, 而且借助教师丰富的教学经验, 引导儿童学习, 避免儿童盲目的玩游戏. 家长也可以随时随地接收到有关作业、考试等消息的推送, 帮助家长了解孩子学习情况, 监督孩子学习.

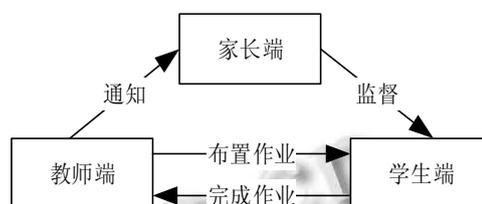


图 1 系统主要功能示意图

2.1.1 教师端功能设计

本系统的教师端是基于 HTML 和 Angularjs 开发的网页端应用. 主要功能如下:

- 1) 教师注册、登陆和账户信息查看功能;
- 2) 创建班级: 录入学生名单, 为学生生成学生端的登陆账号;
- 3) 管理班级: 修改学生信息, 调整班级等;
- 4) 布置作业: 为学生布置作业, 这些作业将会以游戏任务的形式在学生端发布;
- 5) 检查作业: 查看作业的班级完成率, 得分, 排名等;
- 6) 备课: 教师可以使用系统提供的多媒体课件进行备课, 增加课堂趣味性和吸引力.

2.1.2 学生端功能设计

本系统的学生端是基于 Flash 和 Actionscript 3.0 开发的网页游戏应用. 主要有下列功能:

- 1) 登陆: 学生得到教师分发的账号后, 就可以登陆学生端系统;
- 2) 完成作业(完成任务): 教师端布置作业后, 学生端会有相应的任务发布. 学生在游戏系统中需优先完成作业任务;
- 3) 新的游戏关卡解锁: 在任务完成后, 可以解锁新的游戏关卡, 获取课外知识;
- 4) 各类游戏道具、装饰、小宠物, 增加游戏系统趣味性.

2.1.3 家长端功能设计

本系统家长端是基于微信开发的公众号应用. 主要功能如下:

- 1) 接收作业通知: 教师端布置作业后, 会给家长

端发送作业通知;

2) 关联学生账号: 家长可在本端中完成与孩子的绑定;

3) 查看学习的作业完成情况: 家长可在本端上查看孩子作业的各项得分明细.

2.2 系统服务端设计

本系统整体上采用的是浏览器/服务器(Browser/Server, 简称 B/S)体系. B/S 体系与传统的客户端/服务器(Client/Server, 简称 C/S)相比, 有诸多优势. 例如: 维护和升级方式简单、成本低等. 但 B/S 体系也有缺点——应用服务器运行数据负荷较重. 尤其是在面对大量用户访问、高并发请求时, 服务器将面临巨大考验. 为解决高并发 Web 应用系统响应时间过长、系统易崩溃等性能问题, 通常可采用负载均衡与 CDN(Content Delivery Network)、服务器集群技术、增加缓存等技术^[8]. 在本系统中, 使用 Dubbo 分布式集群框架实现负载均衡和服务器集群, 使用 redis 实现数据缓存; 分别从数据库、后台程序代码、Web 应用中间件配置以及服务器负载几个方面对服务器性能进行优化, 提高系统稳定性.

3 系统架构

在本系统中, 后台服务端是最重要的部分, 它为整个系统提供数据、资源及逻辑服务. 后台服务能否

经受高并发访问的考验是整个系统能否稳定运行的关键. 目前大型网站的解决高并发访问方式是采用 SOA(面向服务的体系结构)架构体系^[9]. Dubbo 是一个 SOA 服务治理解决方案, 本系统选择 Dubbo 分布式架构, 并与 SpringMVC 框架整合, 实现数据服务, 数据库采用读写分离和增加缓存的方式提高效率.

系统应用端(教师端、学生端或者家长端)在使用过程中, 都会与系统数据库产生数据交互, 应用端的数据请求将首先发送到后台的控制器, 由控制器选择相应的业务逻辑接口执行. 业务逻辑接口根据应用端的需求与数据库进行交互. 应用 Dubbo 架构以后, 业务逻辑接口即为数据服务提供者(Provider), 控制器作为消费者(Consumer). Dubbo 架构启动时, Provider 将自己提供数据服务的所有接口注册到注册中心(Zookeeper); Consumer 则向注册中心(Zookeeper)订阅应用端所需要的服务. 当应用端需要数据服务并向后台发送请求后, Consumer 会首先接收到应用端的请求, 根据 Zookeeper 提供的服务地址列表, 基于软负载均衡算法, 选一台 Provider 进行调用, 如果调用失败, 再选另一台调用. Provider 根据请求操作缓存和数据库, 将结果返回给 Consumer, 再由 Consumer 返回给应用端. Provider 和 Consumer, 在内存中累计调用次数和调用时间, 定时每分钟发送一次统计数据到监控中心(Monitor)^[10].

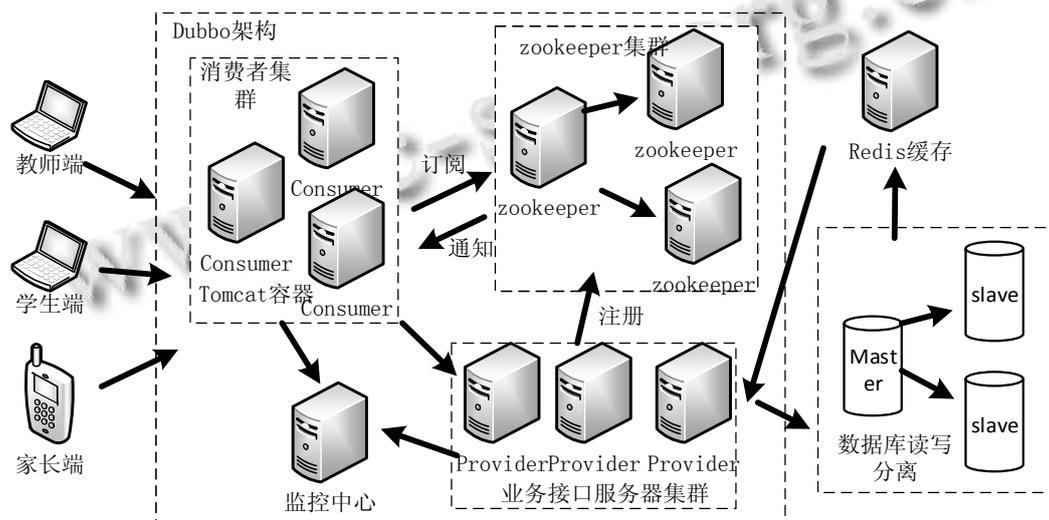


图2 系统架构图

3.1 基于 Dubbo 的后台服务设计

Dubbo 服务端架构包含下列几个部分:

- ① Provider: 暴露服务的提供方;
- ② Consumer: 调用远程服务的消费方;

③ Registry: 服务注册与发现的注册中心;

④ Monitor: 统计服务的调用次调和调用时间的监控中心;

⑤ Container: 服务运行容器.

3.1.1 注册中心 ZooKeeper

ZooKeeper 是一个高性能, 分布式的, 开源分布式应用协调服务. 它提供了简单原始的功能, 分布式应用可以基于它实现更高级的服务, 如同步、配置管理、集群管理、命名空间等. 当服务器启动时, 会在 Zookeeper 的指定目录下创建对应临时节点(这个过程称为“注册”), 当服务器出现故障(无法向 Zookeeper 服务器发送数据包), Zookeeper 会删除临时节点. 当我们的应用端(教师端、学生端或者家长端)需要访问服务时, 需要连接 Zookeeper, 获得指定目录下的临时节点列表, 也就是已经注册的服务器信息, 由对应的服务接口提供服务. 本系统以 Zookeeper 作为注册中心.

3.1.2 服务提供者(Provider)

在本系统中, 服务提供者采用 Hibernate +Spring 框架开发, 主要的后台服务功能有:

1) 账户服务: 提供账号注册、登陆、账号详细信息填写级修改、账号充值等功能;

2) 作业生成以及游戏数据服务: 根据当前教学进度, 生成不同类型的作业(单词拼写类、跟读类、阅读类等), 并存入库;

3) 任务系统服务: 使学生依据任务流程完成作业任务, 保存任务进度, 判断作业任务是否完成等;

4) 排名服务: 根据学生作业完成情况进行排名, 并实时更新排名;

5) 游戏活动相关的服务: 主要完成奖励结算, 奖励兑换, 游戏宠物领养等功能.

这些功能服务在开发过程中, 都对应若干个接口, 系统部署时, 将首先把所有的服务接口注册到 Zookeeper 注册中心, 由 Zookeeper 暴露服务.

3.1.3 服务消费者(Consumer)

服务消费者(Consumer), 采用 SpringMVC 框架开发, 向注册中心(Registry)订阅自己所需的服务, 根据注册中心提供的服务提供者(Provider)地址调用远程服务的消费方. 在本系统中, 消费者根据前端应用的功能需求订阅服务接口.

3.1.4 服务运行容器(Container)

服务提供者(Provider)可以采用 standalone 模式启

动, 也可以使用 Web 容器启动. 由于服务提供者不需要 Tomcat 或 JBoss 等 Web 容器的功能, 如果用 Web 容器去加载服务提供方, 不但增加复杂性, 也浪费资源. 在本系统中, 服务提供者采用 standalone 模式启动. 而服务消费者(Consumer)则需要在 Web 容器中启动, 本系统使用 Tomcat 作为容器.

3.1.5 监控中心(Monitor)

监控中心(Monitor)用来统计服务的调用次调和调用时间等信息, 维护人员可通过这些监控数据掌握各个服务器节点的负载情况, 必要时调整负载均衡算法等. 本系统使用的是 dubbo-monitor 简易监控中心(Simple Monitor). 由于 Simple Monitor 宕机不会影响到 Consumer 和 Provider 之间的调用, 所以用于生产环境不会有风险. 通过配置, 将监控数据(主要是统计数据 and 表图数据)保存到服务器磁盘上, 以便查看.

3.2 数据库设计

数据库瓶颈是高并发系统不可忽视的问题. 数据库出现了瓶颈并不是程序存在逻辑性错误, 而是数据库因为承受了太多的访问后, 数据库无法迅速的做出响应, 严重时候数据库会拒绝进一步操作死锁在那里不能做出任何反应. 要提高系统稳定性, 就必须解决数据库存储问题. 数据库读写分离, 从业务角度而言它是一种粗粒度的数据拆分, 从技术角度而言, 实现手段也相对简单, 因此读写分离是一种低成本解决存储瓶颈的一种手段. 缓存技术也能加快数据库访问响应, 缓存将数据存在内存中, 内存效率是硬盘的几万倍, 若需要的数据已被缓存, 则查询效率大大提高. 本系统的数据库设计就是用了读写分离技术和缓存技术.

3.2.1 数据库读写分离

本系统中, 教师端、学生端和家长端都公用一个数据库. 系统上线以后, 由于系统本身需要收录大量的单词、句子、知识点等信息, 随着业务不断扩展, 数据不断增加, 用户越来越多, 作业系统、任务系统会向数据库中添加越来越多的记录, 数据库的压力也就越来越大, 采用传统的方式, 比如: 数据库的分区、分表或者 SQL 的优化基本已达不到要求, 这个时候就需要采用读写分离策略, 读写分离可以极大地提升数据的读取性能以及整个系统的业务承载能力^[11].

数据库读写分离, 就是一个主数据库(Master), 多个从数据(Slave). Master 库负责数据更新和实时数据

查询, Slave 库当然负责非实时数据查询. 因为在实际的应用中, 数据库都是读多写少(读取数据的频率高, 更新数据的频率相对较少), 而读取数据通常耗时比较长, 占用数据库服务器的 CPU 较多, 从而影响用户体验. 通常把查询从 Master 库中抽取出来, 用多个 Slave 库查询, 使用负载均衡, 减轻每个 Slave 库的查询压力.

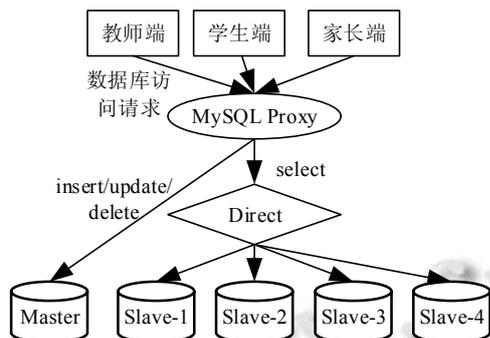


图3 数据库读写分离图

3.2.2 数据库缓存

在本系统中, 如果用户每次进入页面的时候都查询数据库生成页面内容的话, 若访问量非常大, 则系统的对数据库的访问将会非常频繁, 造成数据库负载过大, 网站显示延迟等影响. 缓存技术就是解决此问题的一种方案^[12]. 缓存(Cache)是一种用空间换取时间的技术, 存在于计算机中很多地方, 用来将一些慢速设备中的常用数据保存在快速设备中, 取数据的时候直接从快速设备中取.

Redis(Remote Dictionary Server)是一个基于内存的高性能 key/value 数据库, 与 Memcached 一样, 为了保证效率, 数据都是缓存在内存中. 最大的区别是 Redis 会周期性的把更新的数据写入磁盘或者把修改操作写入追加的记录文件, 并且在此基础上实现了 master/slave(主从)同步. Redis 的出现, 是对关系数据库很好的补充. 本系统使用 Redis 作为 MySQL 的缓存, 为系统提供数据支持.

4 结语

本文提出了在线儿童英语学习平台的设计和实现, 以游戏吸引儿童注意力, 让儿童在玩游戏的过程中潜移默化地进行学习. 教师端、学生端和家長端, 三端互联. 教育、学习、监督一体化. 本系统将教育游戏的吸引力与教师教学的指引相结合, 既提高了儿童学习英语的自主性, 又为儿童提供了与教学大纲相匹配的学习内容, 让儿童不再盲目学习. 本系统与北京某公司合作实现, 由该公司向学校推广. 目前, 本系统已成功进入西安市几所小学, 作为其教师备课、学生完成作业的指定系统. 2016年1月寒假开始时, 系统切合时宜的推出了一系列寒假活动, 带来了用户量的激增. 在此期间, 后台业务逻辑接口仍然提供高效稳定的服务, 这都得益于后台服务端所采用的分布式集群框架以及数据库读写分离等优化.

参考文献

- 1 朱永海, 韩锡斌, 杨娟, 程建钢. 高等教育借助在线发展已成不可逆转的趋势. 清华大学教育研究, 2014, 35(4): 92-100.
- 2 MOOC.MOOC 学院. <http://mooc.guokr.com/>.
- 3 姜艳, 陈爱东. 基于网络的儿童游戏化虚拟学习社区构建分析. 中国教育信息化, 2009, (8): 72-74.
- 4 网娃. 网娃乐园. <http://www.gogo517.com/>.
- 5 企鹅俱乐部. Clubpenguin. <http://www.clubpenguin.com/>.
- 6 摩尔庄园. 摩尔庄园. <http://mole.61.com/>.
- 7 阿里巴巴. Dubbo 用户指南. <http://dubbo.io/User+Guide-zh.htm>.
- 8 包立辉, 黄彦飞. 高并发网站的架构研究及解决方案. 计算机科学, 2012, (Z2): 184-187.
- 9 Mark D.Hansen(美). 使用 Java Web 服务构建 SOA. 北京: 电子工业出版社, 2009.
- 10 谢璐俊, 杨鹤彪. 基于 Dubbox 的分布式服务架构设计与实现. 软件导刊, 2016, 15(5): 13-16.
- 11 沙光华, 陈泳, 张长江. 读写分离技术在运营支撑系统中的应用. 计算机工程与应用, 2015, 51(12): 107-110.
- 12 曾超宇, 李金香. Redis 在高速缓存系统中的应用. 微型机与应用, 2013, 32(12): 11-13.