

# A Solution of Collaborative Experiment-Course Management System Based on Cloudy Platform

Xingying Li

Wuhan Media and Communications College of Huazhong Normal University, Wuhan Hubei  
Email: [8705541@qq.com](mailto:8705541@qq.com)

Received: Jul. 29<sup>th</sup>, 2015; accepted: Aug. 13<sup>th</sup>, 2015; published: Aug. 19<sup>th</sup>, 2015

Copyright © 2015 by author and Hans Publishers Inc.  
This work is licensed under the Creative Commons Attribution International License (CC BY).  
<http://creativecommons.org/licenses/by/4.0/>



Open Access

---

## Abstract

Experimental teaching is the implementation focus of colleges and universities. To meet the requirements of the experimental teaching, I design a set of management system of the remote experiment. In this way, teachers will have a better method for routine experiment teaching, students are more effective to participate in the experiment and have better communication and interaction with teachers, and at the same the teaching departments can process supervision and regulation of experimental teaching. Based on the remote experiment management system design, and combined with cloud platform technology for further in-depth analysis and extension, the application of the system on the cloud platform can be strengthened, and the late operational cost can be reduced.

## Keywords

Cloudy-Platform, Experiment-Course, Collaborative, OpenStack Zookeeper

---

# 基于云平台的实验课协同化管理系统

李兴莹

华中师范大学武汉传媒学院, 湖北 武汉  
Email: [8705541@qq.com](mailto:8705541@qq.com)

收稿日期: 2015年7月29日; 录用日期: 2015年8月13日; 发布日期: 2015年8月19日

文章引用: 李兴莹. 基于云平台的实验课协同化管理系统[J]. 计算机科学与应用, 2015, 5(7): 264-270.  
<http://dx.doi.org/10.12677/csa.2015.57034>

## 摘要

实验教学是各大高校的实施重点，为满足实验教学需要，让教师有更好的手段进行日常实验教学，学生能更有效的参与到实验环节，并与教师有更充分的交流和互动，同时教学部门也能对实验教学的流程进行监督和监管，设计与研发了一套远程实验课的管理系统，在该远程实验课管理系统的设计基础之上，结合云平台技术进行了进一步的深入分析和扩展，强化了这套系统在云平台上的应用，并能减少后期的运维成本。

## 关键词

云平台，实验课，协同化，OpenStack Zookeeper

## 1. 引言

随着社会的需要和教育部所提倡的应用型复合型人才的培养，各大高校均加强了实验课程的教学力度，投入了大量的资金进行硬件的升级、改造或软件的普及，对实验课程的教学及效果非常的重视，包括校企合作也有一部分是在学校的实验室里进行展开的实训。在这一过程中，如何对实验教学环节进行整体的管理，包括实验教学任务的设计、教师与学生的互动、教学部门的监管，都成为一个难点，如何进行实验课的协同化管理逐步在实验教学环节中凸显出来。

随着 IT 技术手段和领域的不断拓展，从私有云到公有云，再到现在的混合云，云计算发展得非常迅速，云化应用已成为不可扭转的大趋势[1]。鉴于云基础平台相关的技术已日趋完善，结合目前的实验课教学需要，基于云平台的实验课协同化管理系统应运而生。

## 2. 总体方案

整个系统分为三层：云基础平台层、云应用层和云系统管控层，其结构如图 1，系统逻辑架构图。



Figure 1. Logical structure of system

图 1. 系统逻辑架构

## 2.1. 云基础平台层

云基础平台层是该实验课管理系统的基础层，为云应用层提供所需的 IT 基础设施，包括：计算、存储和网络等资源。采用 OpenStack 框架[2]，主要是实现该平台的管理和使用，就像是云平台的底层操作系统，它是安装在真实的物理机上的，然后会虚拟出 N 个虚拟机，利用这样的实现非常便于管理和使用。在传统平台中，计算、存储与网络这些资源被某些程序或应用独占，那些高负荷的应用得不到所需要的全部资源，而低负荷的应用又闲置了大量资源，造成资源使用不均衡。现在有了云基础平台之后，这些资源在云基础平台中被池化[3]，根据云应用的需要可以进行动态分配，这样就提高了资源的利用效率，并有效地节省了硬件的投入成本。

该系统的云基础平台层基于 OpenStack 构建。OpenStack 是一个构建公有云或私有云的开源软件项目，它包含了一系列子项目，涵盖控制管理、存储服务、镜像服务、网络服务等等。本系统中采用 KVM 作为虚拟化管理程序，KVM 可以与 OpenStack 进行集成。KVM 作为 Linux 系统的组件之一，可以很容易部署到 Linux 服务器上，因此选用 Linux 服务器作为云平台宿主物理服务器。

为了实现应用的快速部署，将应用环境制作成 Docker 镜像，在 OpenStack 中管理 Docker。Docker 是一个开源的应用容器引擎，可以在这个容器中打包所有与实验平台有关的软件。在任何需要的时候，将这个 Docker 容器镜像发布到 KVM 中。使用者不再需要自己安装运行环境，只需要做一些简单的设置即可。

## 2.2. 云应用层

该系统的云应用层是基于云基础平台层所提供的 IT 基础设施进行部署的。在基础平台硬件环境之上，安装常用软件，比如操作系统、数据库等。这样在云应用层，我们实现了实验课协同化管理系统云化应用。在这一层，系统向用户提供实际的软件功能，应用了 JAVA 相关技术实现实验课协同化的视频、音频、留言等。

这一层实现的应用包括：基础数据管理系统、实验课远程教学系统、虚拟实验室、实验报告管理系统。

## 2.3. 云系统管控层

云系统管控层，负责监控云基础平台的运行状态，并能实现对云应用层的自动化部署、配置和监控。在实验课管理系统中通过这一管控层可以方便快速的查看到系统有没有问题：磁盘有没有用完，CPU 是不是占用太高，网络通不通畅等。

在基于云平台的环境下，根据不同应用需求，会有不同的部署和配置。对于运维工作来讲，没有自动化工具来配合完成的话，那工作量是巨大的。比如 Facebook 数据中心的运维人员每人需要管理 2 万台服务器。云系统管控层就是通过自动化手段，对云平台上的服务器和应用进行配置和监控，及时跟踪并解决运行中出现的问题。

管控行为包括两个层面，一个是虚拟化硬件层面，另一个是应用层面。OpenStack 本身提供了 Ceilometer 组件，可用于监虚拟机的 CPU、磁盘、网络，并具有扩展监控能力。Ceilometer 是通过部署在计算节点上的 Compute 服务，来获取监控数据的。Ceilometer 的默认功能已基本实现对虚拟化硬件层面的监控。此框架如图 2 所示，Ceilometer 框架图。

计算节点上的 agent 服务负责采集信息，通过 RPC、UDP、File 三种方式将采集的信息 publish 给 collector。其中，RPC 是将采集的信息以 payload 方式发布到消息队列，collector 服务通过监听对应的 queue 来收集这些信息，并保存到存储介质中；UDP 通过 socket 创建一个 UDP 数据通道，然后 collector 通过

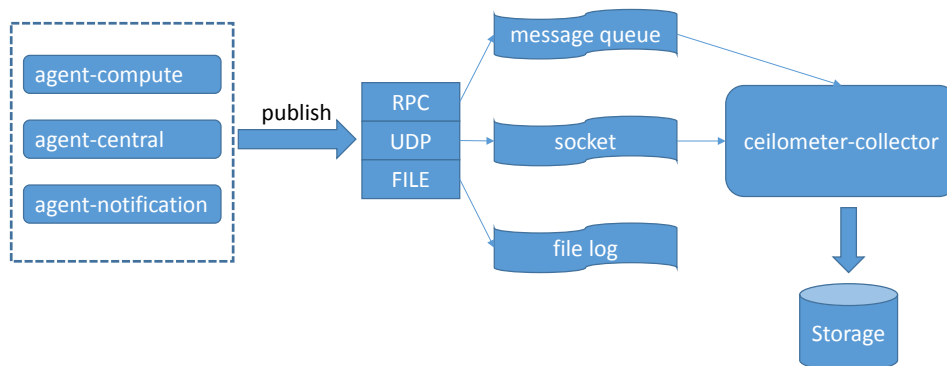


Figure 2. Framework of ceilometer

图 2. Ceilometer 的处理框架

bind 这个 socket 来接收数据，并保存到存储介质中；File 方式比较直接，就是将采集的数据以 filelog 的方式写入 log 文件中。另外，通过对 Ceilometer 的扩展，还可以增加应用层的接口，实现对应用层的监控。自动化部署和配置基于 Puppet 实现[4]。Puppet 是采用 Master-Slave 架构的集中配置管理系统，可管理软件包的部署和配置。Puppet 相对于传统的手工维护，要稳定得多。Puppet 要实现的目标就是使我们能集中于要管理的目标，而不是细节，而把繁琐的过程交给 Puppet 来完成。

通过这三层的相互关联与应用，该系统在实施过程中能到较好的应用。下面详细介绍一下本系统的应用层的实现。

### 3. 应用层实现模型

整个应用层主要有以下部分实现：

基础应用组件，包括实现本系统的公共服务和部分通用应用程序(实现教师管理、学生管理、班级管理、课程管理、系统参数管理、实验报告管理等通用功能)。

公共服务部分，主要包括基础数据的存储和访问服务。

基础数据是指所有子系统都可能需要访问的数据。这些数据的访问量都比较大，但数据的变化很少。比如系统的公共参数、人员、物资等基础信息。由于云应用的特殊性，在服务负载上升时，它必须能实现从单节点服务到多节点服务的平滑迁移。如果服务端点实现了多节点，而数据存储和访问服务仍然只是单节点的话，那么云应用的性能就会大打折扣。而数据服务就成了整个系统的瓶颈。因此不能直接采用传统的 RDBMS 来实现本系统的数据存储和访问服务。该系统中应用分布式数据服务的模型时采用了 ZooKeeper 作为底层框架。ZooKeeper 是一个分布式的，开放源码的分布式应用程序协调服务，其控制方式见图 3 所示。

本系统中，会有多台数据服务器，那么客户端怎么知道应该选择哪一台数据服务器呢。为了解决这个问题，需要在这些数据服务器中设置一台为 Leader (主数据服务器)，它负责接收客户端的请求，然后再根据客户端的请求信息，将客户端请求分配到某个指定的数据服务器上。在这些服务器中，只会有一台服务器是 Leader。在系统启动时，它们都参与 Leader 的选举过程。它们都在 ZooKeeper 中申请一个服务节点序号，并且这个序号值是从小逐渐增大的[5]。所以有一种简单的算法就是选择序号值最小的那个服务节点作为 Leader。Leader 在所有数据服务节点中充当管理员的角色。它需要实现以下功能：

#### 1) 数据同步

Leader 负责将所有数据服务节点的数据同步。一般来讲，Leader 上的数据是最多的。因此 Leader 会定时向其他数据节点同步自己的数据。

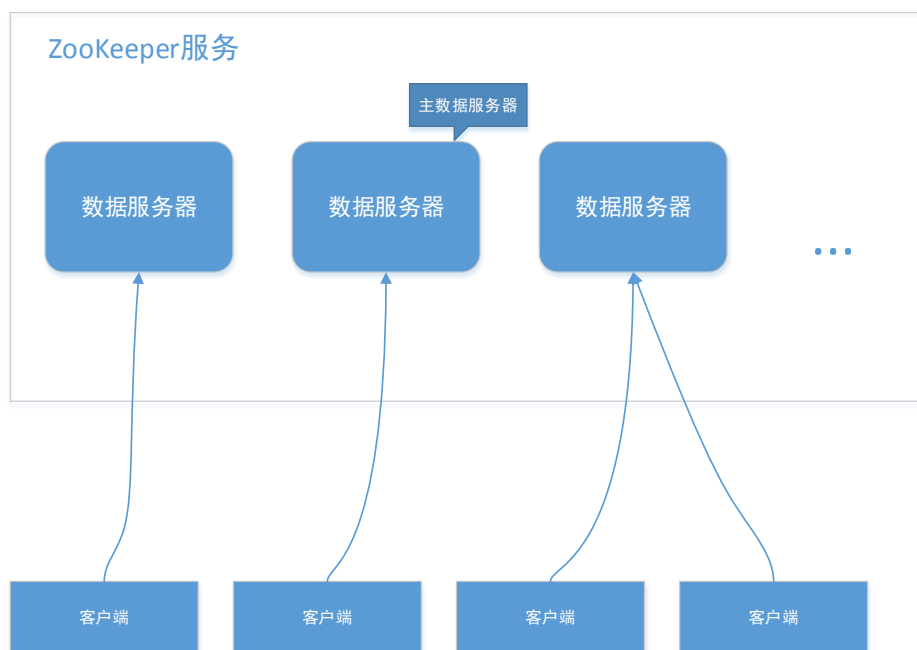


Figure 3. Distributed data access and control of multi-service nodes

图 3. 多服务节点的分布式数据访问与控制

## 2) 负载均衡

Leader 中有一张所有数据节点的负载记录表。每次有数据节点响应服务请求时，Leader 会将请求记录在表中。当服务节点结束响应时，Leader 再从负载记录中清除相应记录。因此 Leader 根据负载记录，可判断哪些服务节点的压力过大，并尽可能将客户端请求分散到多个节点上，以提高系统的整体性能。所有客户端在需要访问数据服务时，首先向 Leader 发出请求。Leader 收到客户端的请求后，再根据以上策略，决定转发到某台目标数据服务器上，由目标数据服务器为客户端提供具体数据服务[6]。

通用应用部分的作用是为配合远程在线教学的需求，对教学资源 and 成果进行管理，比如在线填写和评审实验报告、教师分发实验资料、教学要求、在线答疑等等，实现无纸化教学，符合绿色环保的要求。其整个架构见图 4 所示。

这些应用是基于 Struts + Spring + iBatis 实现。

基础应用组件中采用 Struts 以 MVC 模式实现了 WEB 模块。这种模式的设计方式减弱了业务逻辑接口和数据接口之间的耦合[7]，以及让画面效果更富于变化。Spring 应用在本系统中，主要用于实例化后台服务，管理数据库事务等。Spring 是一个开源的基于 IoC (Inversion of Control, 反向控制)和 AOP 的构架的 Java 框架,采用 IoC 使得可以很容易的实现 bean 的装配,提供了简洁的 AOP 并据此实现 Transaction Management。

远程实验课组件,实现基于 Web 的实验课程在线教学与交互。这部分是基于 Flex 和 J2EE 技术实现,可实现白板、语音、视频、虚拟在线实验等功能。可实现教师与学生一对多或一对一的虚拟面对面教学,比传统远程教学方式能显著提高体验和教学效率。Flex 是基于 Flash 平台的富客户端技术[8],它比传统 HTML 应用更具表现力,可以展现更多的内容(图片、语音、视频等),接近 ActiveX 控件的效果。但由于 Flex 是基于 Flash 实现的,它比 ActiveX 对客户端环境的要求更低,兼容性更强。只要客户端安装了 Flash 插件,它就可以正常运行。基于 Flex 和 RSTP 流媒体传输协议,我们实现了在线教学功能(白板、语音、视频)。这能达到传统远程教学无法比拟的效果。虽然它在性能上可能还比不上某些商业系统,但这套方

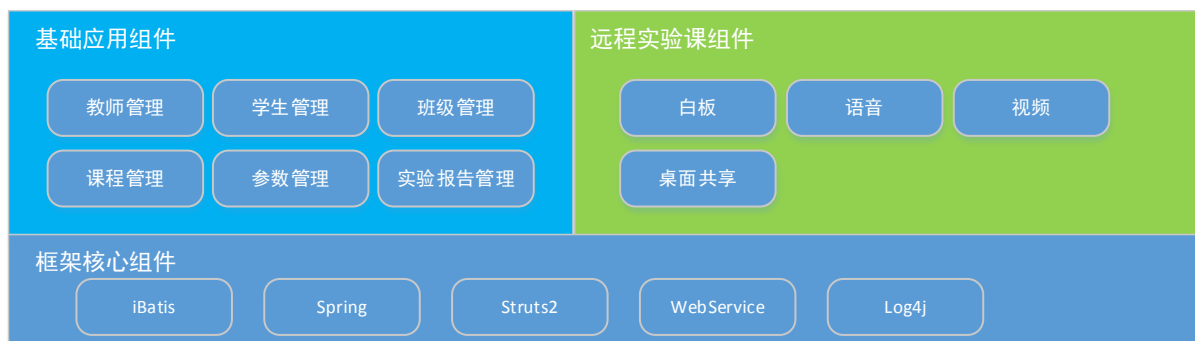


Figure 4. Technical Framework of application layer

图 4. 应用层的技术架构

案完全基于开源技术实现,没有采用任何商业软件。另外基于 Flex 的实现方案,使得终端的兼容性更强,支持 PC、平板、手机等多种终端设备上的浏览器。

在具体实现上,白板与语音、视频稍有区别。语音和视频,在传输过程中直接采用相应的有损二进制压缩方式(语音和视频都采用 3gp 格式压缩),压缩后再以广播方式传输给各个客户端。即使是采用了最好的压缩算法,由于是采用广播模式,当客户端很多时,对系统和网格的压力还是相当大的。

因此在白板的实现中,为了提交传输效率和画面质量,不是记录屏幕的视频数据,而是直接记录了白板使用者的操作轨迹(描点)。然后将轨迹同步给各个客户端,由客户端根据操作者的轨迹,在本机屏幕上重画图像。

与 Flex 相匹配的服务端,采用基于 RESTful Webservice 的技术实现。REST Webservice 的主要特征之一是以遵循 RFC 2616 协议的方式显式使用 HTTP 方法。在本系统中,使用 JSON 数据格式作为接口的数据传输格式。与普通 HTML 数据相比,REST Webservice 的 HTTP 头部不一样:

*HTTP/1.1 200 OK*

*Server: xLightweb/2.6*

*Content-Length: 263*

*Content-Type: application/json; charset=utf-8*

而 HTML 请求的 HTTP 头部则是:

*HTTP/1.1 200 OK*

*Server: xLightweb/2.6*

*Content-Length: 26*

*Content-Type: application/x-www-form-urlencoded; charset=utf-8*

最大的区别是 Content-Type。另外在传输数据时,HTML 请求是返回符合 HTML 格式规范的数据,而 REST Webservice 则返回 JSON 格式文本数据。

#### 4. 后续发展

该实验课协同化管理系统目前已经初步实现了基于文字、语音、视频、桌面共享技术等功能,以及在实验教学中实现了白板、一对多授课功能。对某些实践性要求比较高的实验(比如网络实验),这套方案的教学效果有限。下一步的设计与开发中,该系统会进一步进行完善,设计基于虚拟 3D 技术的在线实验室方案,并基于这套方案再实现相关的课件,使学生可以身临其境地在虚拟环境中进行实验,同时设计相关的手机 APP 客户端,方便学生或教师随时进行提问和答疑。



另外本系统的云化应用程度还比较低，目前只是能把应用部署到云上，但对用户数的支持仍有限。在改进中会考虑将核心框架改写成分布式服务体系，使得系统可以根据用户访问量进行动态扩容，这样系统的弹性会更强。

## 5. 结束语

本项目的设计与开发的目的是要完成一个实验课程的协同化管理系统，能对实验课整个流程进行动态的跟踪和记录，让学生能随时提问，能看到老师的示范操作；让教师能随时答疑，并批改学生的作业；让管理部门能跟踪和监督。因此该系统有一定的实用价值。

### 1) 创新之处

基于云平台的应用，是未来的发展方向。怎样发挥云的潜力，简化应用的开发、部署和维护过程，在本系统中有一定的探索。采用 OpenStack 等一系列开源框架或工具搭建云平台，以此简化了云平台的部署和维护过程。同时在开发过程中，为了解决分布式数据的传输和存储，也进行了一系列尝试。比如基于 ZooKeeper 的分布式数据存储和访问机制，这些可以应用到更多的业务场景。

通过实现这一整套应用，本系统提供了先进的实验问题实时协作处理平台。学生可以在本平台上提出问题，教师可以借助网络实时手段(包括文字、语音、视频等)，对学生的问题进行协助解答，也可以使用手机 APP 应用。通过调研可以发现这样的系统目前在实验教学中还没有进行应用。而现有的一些管理系统功能都比较单一。

### 2) 实施范围及受益对象

该套系统成功实施后，涉及到的角色有实验室管理人员、参与实验的学生及老师，学校相关职能部门。各角色都能在整个实验中动态的进行记录和跟踪，能规范实验教学，加强学生的互动性和学习的自觉性。

### 3) 不足之处

本系统中分布式应用的主体框架是基于 ZooKeeper 构建，通过它很方便地实现了各个服务节点之间的协调管理工作。但 ZooKeeper 的部署比较复杂，而且还不能配合云平台实现服务节点的动态调整。未来我计划实现基于一套新的分布式服务框架，简化部署过程，支持服务节点的动态调整，可任意增加或删除节点，而无须重新启动系统。

该系统还在实践基础上不断探索和完善，以期能真正帮助教师和学生改善教学效果，最终能将系统全面推广到教学实践中。

## 参考文献 (References)

- [1] 范平 (2012) 公有云私有云都是过客 混合云才是王道. <http://server.zol.com.cn/276/2761222.html>
- [2] 高贵升 (2012) 基于 OpenStack 的计算云的研究与实现.
- [3] 赵新芬 (2013) 典型云计算平台与应用教程. 电子工业出版社, 北京.
- [4] 黄凯 (2014) OpenStack 实战指南. 机械工业出版社, 北京.
- [5] 杨欢 (2014) 云数据中心构建实战: 核心技术、运维管理、安全与高可用. 机械工业出版社, 北京.
- [6] Benjamin, E. (2015) 大规模 java 平台虚拟化与调优. 机械工业出版社, 北京.
- [7] 缪勇 (2015) Struts2 Spring3 Hibernate 框架技术精讲与整合案例. 清华大学出版社, 北京.
- [8] 钱钰, 陈志云 (2009) 基于 Flex 的 RIA 技术在教学软件中的应用. 计算机与数字工程.