**前言**

本文是外包学生管理系统的详细架构设计文档，用于指导系统的后续开发和运维。

**修订历史**

|  |  |  |  |
| --- | --- | --- | --- |
| 版本号 | 变更内容 | 修订者 | 日期 |
| 1.0 | 创建全文 | 焦俊文 | 20220731 |
|  |  |  |  |
|  |  |  |  |

**词汇表**

无

1. **业务背景**

随着互联网的发展，技术越来越成熟，成本越来越低廉，在学校规模发展愈来愈大的趋势下，学校的管理面临以下几个痛点：

* 选课流程：采取线下选课的方式，自由选课需要去线下排队投票成本太高。
* 考试流程：需要老师提前进行预定以及通知对应学生和监考老师。
* 入学流程：需要到档案处办理以及核对学生信息真伪。
* 老师日常管理：需要老师线下到信息处进行处理。
* 学校成果管理：和外校其他系统进行成果交流困难，需要使用线下的方式整理材料。

所以需要把师生信息进行电子化，学校的课程考试管理信息化，日常管理流程化。因此，推出一个可供在校员工和学生可以登录的一个线上管理系统势在必行。

**2. 约束和限制**

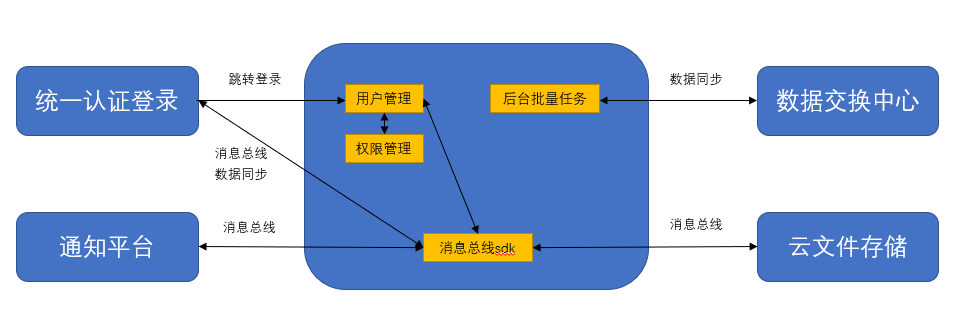
1. 必须在指定工期前完成。
2. 成本不能超过200W。
3. 数据库需要保证数据不丢失。
4. 需要和其他外校/本校系统进行良好交互。
5. 符合当地政府的信息安全要求。
6. 峰值tps保证2000，日常500左右即可。
7. 前后端分离。

**3. 总体架构**

1. 系统边界黑盒图



1. 系统边界白盒图



**3.1 架构分析***]*

* + 1. 高可用

由于客户要求，对于系统的数据存储需要做到保存完备，所以系统采用了Mysql集群，并且采用了主备模式，数据复制方式采用同步复制，保证一致性。

同时为了前后端分离，分离出了前端服务器，用于处理前端请求以及各种平台（桌面端和手机端）的前端页面请求。

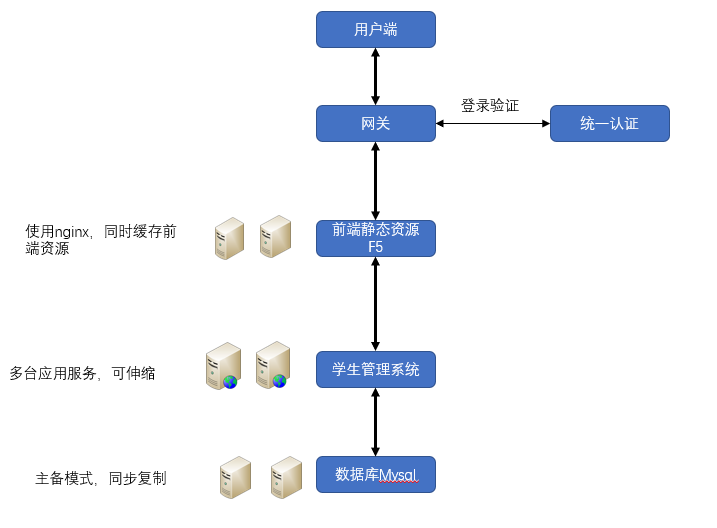
应用采用多台服务器，通过nginx反向代理分发任务处理请求。如果一台出现crash，也可以有其他服务器提供服务。

* + 1. 高性能

性能要求是峰值2000tps，日常500tps，2台应用服务器。同时业务对于选课高峰时期进行错开即可。

**3.2 总体架构**

3.2.1总体架构图

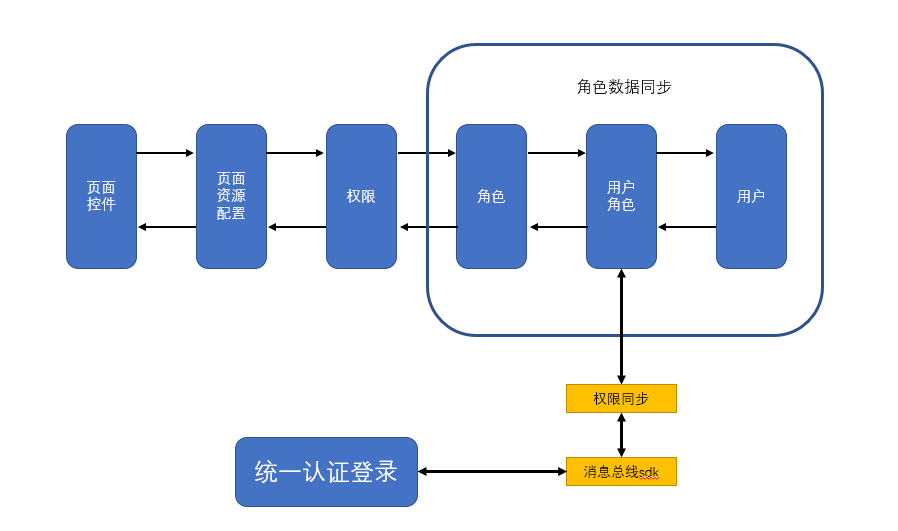


* 应用采用集群模式，默认是2台服务器，可以弹性伸缩，通过前端f5服务器一致性hash分发请求。
* 数据库采用mysql主备模式，数据复制方式采用同步复制，保证一致性。
* F5服务器默认是2台服务器，稳定提供服务。

**4. 详细设计**

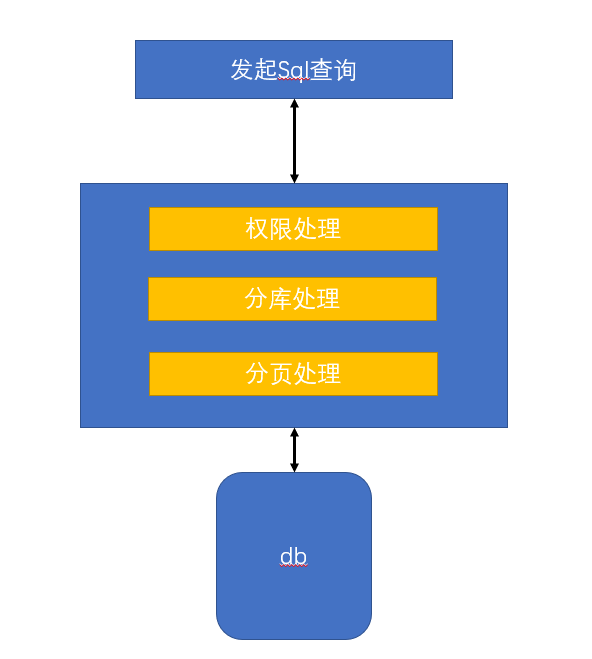
**4.1 核心功能**

4.1.1页面权限处理流程



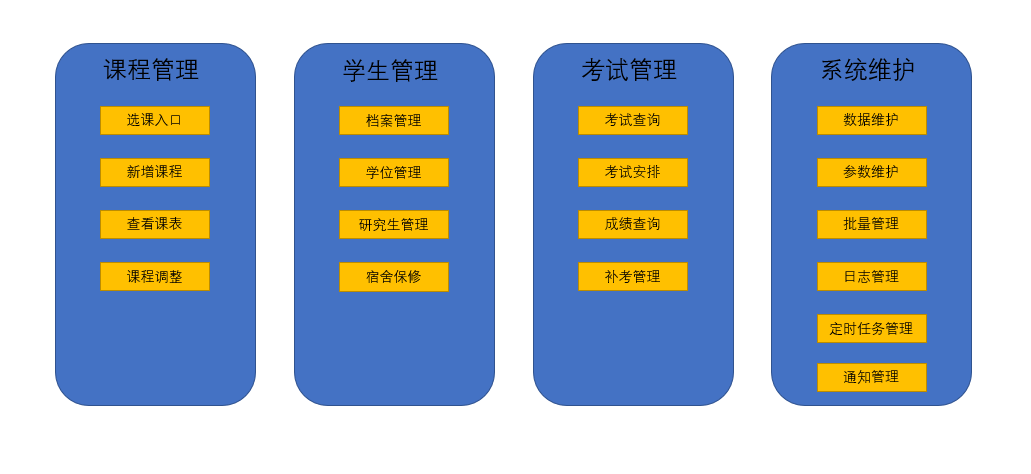
使用6张常用表来管理页面资源，具体角色权限配置则交给统一认证系统管理，学生数据在开学的时候由后台批量导入。

4.1.2数据权限处理流程



在sql查询的时候，使用拦截器进行sql处理，筛选数据。

4.1.3主要功能



4.1.4系统架构：



**4.2 关键设计**

1. 由于数据权限集中管理，所以把权限数据改为了从统一认证系统进行数据同步，同时学院和学校的机构消息也进行同步。
2. 关于数据量问题，由于学校规模一年会有5000左右的学生，所以业务数据量不大，暂时还不用分表，但是历史记录采取超过2年备份处理。为了提高查询效率，学生id采取（年份+学院编号+学号）来进行存储。
3. 由于需要和校内其他系统进行实时交互，比如文件存储平台，用来保存学生的图片消息。因此开发了对应针对校内消息总线的模块作为防腐层，进行解耦。
4. MySQL采取的是主备模式，并且是同步复制，所以对MySQL的磁盘采取了ssd存储配置。
5. 服务器采取2台服务器的集群，并且支持伸缩扩展，同时针对高峰业务进行错峰处理（比如学生晚间选课）。
6. 安全通过学校的统一认证jwt+outh2认证，同时上线前进行安全性扫描。

**4.3 设计规范**

1. 前端使用vue进行开发。
2. 后端使用springboot进行开发
3. MySQL使用innodb存储引擎。
4. 前后台交互使用http rest风格协议，消息总线交易对消息总线对应格式协议进行适配。
5. 系统暂时不进行划分，模块合并到一个war开发，但是相关配置的配置参数配置在数据库里
6. 服务器采取独立tomcat，并且打开jndi配置，方便修改对应数据源。连接池配置在每台服务器500左右即可。

**5. 质量设计**

5.1、可测试性

5.1.1功能需要由单元测试，上线前需要测试覆盖率达到80%。同时接口应该支持幂等性，可以直接通过postman等工具进行测试。

5.2、可维护性

5.2.1提供了后台系统维护模块，可以进行sql注入的数据维护以及系统可配置化参数的维护。如果服务器资源告警则由运维发布事件单通知it。

5.3、可观测性

5.3.1提供了对应的日志下载模块，采用最新的log版本，后台系统维护模块可以直接下载日志进行数据的查看情况，或者采取其他监控读取服务日志。

5.4、成本

5.4.1成本方面，基础需要6台服务器左右，同时需要开发以及后期运维费用。能够控制在200以下。

**6. 演进规划**

6.1、学生管理系统一期

实现学生管理系统模块的全部功能，并且成功上线交付。

6.2、学生管理系统二期

实现学生管理系统的上云。

6.3、学生管理系统二期

进行服务器拆分，拆分后端为微服务。