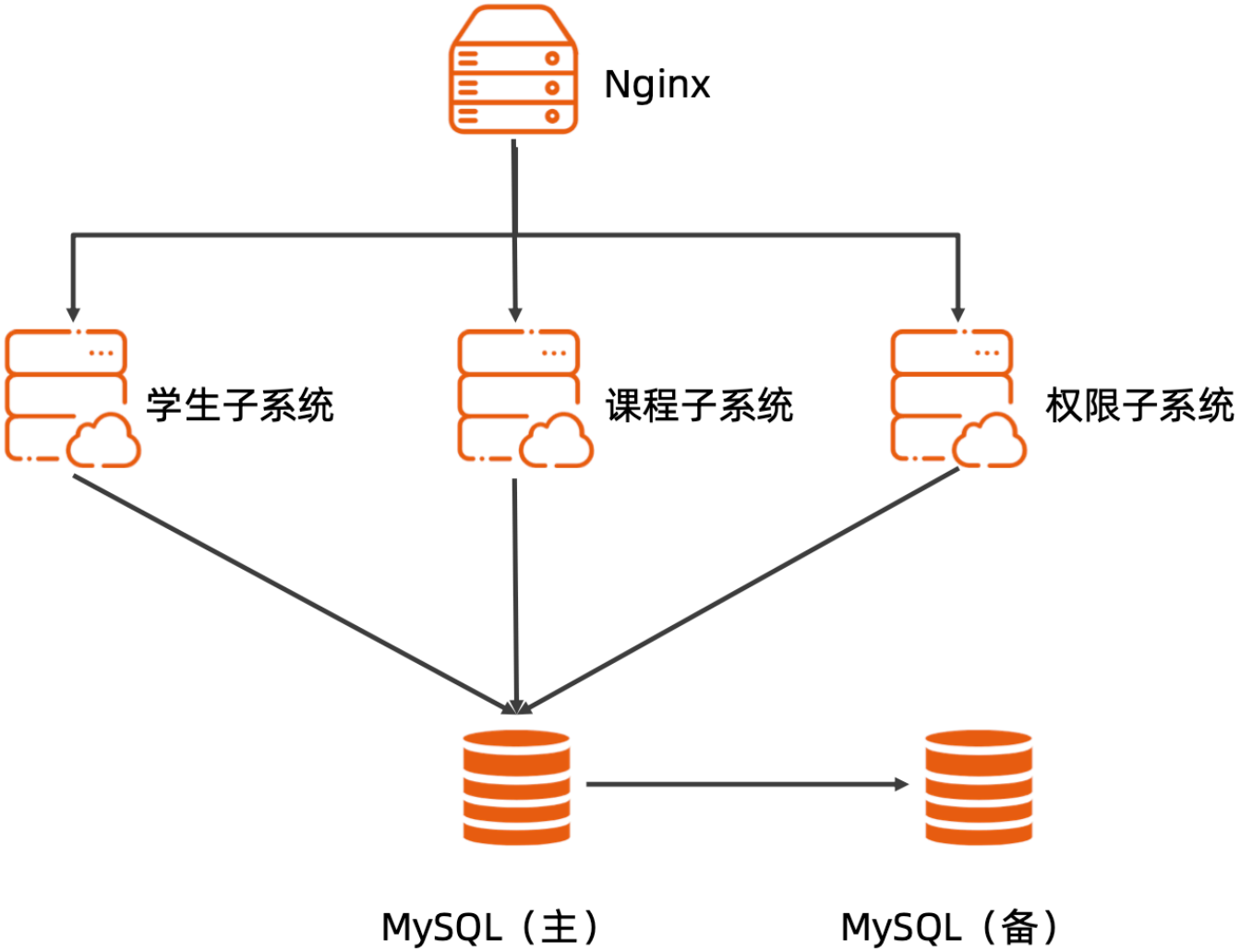
1. 业务背景

随着学校规模不断扩大，学生数量的增加，需要处理的信息也日趋增大，所以，需要构建一个学生管理系统来应对这个问题。该系统主要应用于学校学生信息管理，总体任务是实现学生信息管理的系统化、规范 化和自动化，其主要任务是管理学生相关信息，如学籍、课程、成绩、奖惩。具体可分为学生管理、课程管理、考试管理和权限管理。

2. 约束和限制

1. 必须在 2022.06.30 号完成
2. 该系统能够支撑管理3万个学生
3. 成本不超过1000万
4. 数据库采用MySQL
5. 团队有6个人，都会 Java，有一个是C/C++高手

3. 总体架构设计



总体架构分为3层：

1. Nginx：负责请求接入并router到相应的子系统上
2. 子系统层
    1. 学生子系统
    2. 课程子系统：包含考试
    3. 权限子系统：包含学生、老师等
3. 存储层：MySQL主备架构

4. 详细架构设计

4.1 核心功能

4.1.1 创建学校

DBA 创建数据库 ---> 运维创建域名，添加 Nginx 配置 ---> 运营往数据库中添加学校的信息（包括创建年级、备选课程等）

4.1.2 用户注册

4.1.3 选课功能

学生可以在线对自己的课程体系进行选择，相对应的课程选择功能类比。

Nginx ---> 课程子系统

4.2 关键设计

1. 高可用：数据不能全部丢失，数据要高可用
    1. 消息存储在一主一备 MySQL 中，主备MySQL 服务器之间复制消息以保证消息存储高可用。
    2. 如果主备间出现复制延迟，恰好此时 MySQL 主服务器宕机导致数据无法恢复，则部分消息会永久丢失，这种情况不做针对性设计，DBA 需要对主备间的复制延迟进行监控，当复制延迟超过 30 秒的时候需要及时告

警并进行处理。
    3. 主服务器提供消息读写操作，从服务器只提供消息读取操作。
2. 可扩展：业务需求比较复杂

4.3 设计规范

1. 子系统用 Spring Boot + Netty 开发
2. MySQL 使用 Innodb 存储引擎

5. 架构质量设计

1. 可运维性：在DB中增加/减少学生，增加/减少课程等，可以引入一个运维子系统

6. 架构演进规划

1. 实现现在的这个总体架构
2. 将所有子系统放在一个业务服务器中，部署多台服务器提升整个系统性能。单个业务服务器里面分成3个模块（学生管理、课程管理、权限管理）