

基本信息

- 姓名: 王贤
- 手机号: 18335103278
- 邮箱: xianger94@gmail.com
- 院校: 太原科技大学网络工程/16届
- github: <https://github.com/shfshanyue>
- 博客: <https://shanyue.tech>
- 线上简历: <https://github.com/shfshanyue/AboutMe/tree/develop>

工作经历

- 2016/07 - 至今 **智课教育**
| 前端, 爬虫, 后端, devops, ETL
- 2015/12 - 2016/03 **豆瓣**
| 前端

简述

在公司负责一个项目组的日常业务开发, 但也参加过公司几乎所有业务线前后端项目的开发。另外也会使用 scrapy 做一些爬虫。另外我拥有公司全部业务生产环境和数据库的权限, 也会做一些数据处理 (ETL) 的工作。另外还解决一些业务与技术上的问题, 包括但不限于:

1. 考试系统班级中对所有同学的考试报告进行批量打包下载 (因此也独立一人做了关于 url2pdf 的服务)
2. 使用 graphql 对教育云后端进行重构并渐进式替代老后端
3. 容器化前后端项目, 并添加 CI/CD 集成到 k8s 进行部署
4. 日志与异常上报的优化, 添加 requestId, 快速定位问题以及优化项目
5. 监控 cpu/memory, 多次解决 OOM 问题, 并优化监控系统的警告 (如1min内存超50%进行邮件通知)

对自己而言, 会折腾一些服务器的事情: **当我有一台服务器时我做了什么**, 另外我的博客 <https://shanyue.tech> 也会记录自己一些工作生活上的思考。

另外个人 github 也有一些系列笔记:

- **个人服务器运维指南**
- **k8s实践**
- **使用graphql构建web应用**

项目经验

一个关于 graphql server 的脚手架

简介

- 项目地址: <https://github.com/shfshanyue/apollo-server-starter>

在公司生产环境使用 graphql 后，总结了一套关于 graphql 的脚手架，同时也需要公司集成一些较好的基础服务，比如 k8s, consul, sentry, elk 等

技术栈

- GraphQL.js, apollo-server, koa, DataLoader — API 层
- PostgreSQL, Redis, ioredis, sequelize, lru-cache -- 存储
- TypeScript, sequelize-typescript -- ts 支持
- Joi, consul, node-consul, winston, sentry -- 校验, 配置, 日志与报警
- Docker, docker-compose, gitlab-CI, traefik, kubernetes -- 部署

教育云系统

- 2017/01 - 2017/07 前端
- 2017/07 - 2018/08 全栈
- 2018/08 - 2019/04 后端 (Node) (leader 4人)

简介

教育云是一个服务于各大高校的一个 toB 的项目，主要业务为考试系统，课程服务，以及写作的 AI 批改。负责项目的正常迭代，开发以及优化

技术栈

凡有列出个人必有涉及

前端: react typescript mpvue apollo-client 运维: k8s gitlab-ci elk sentry prometheus 后端: koa apollo-server websocket postgres redis

关于 graphql 的部分写了一个脚手架: <https://github.com/shfshanyue/apollo-server-starter>

日常工作

- 维护和优化前后端项目，负责前后端项目的任务分配，代码审核以及上线，解决日常反馈问题
- 监控 sentry 系统，及时解决线上问题
- 监控 elk，及时发现慢查询，慢请求，不合理的 SQL (如一条 api 对应 100 条 SQL) 并及时解决问题
- 与产品以及商务开会，确定需求

详细

- 负责并独立开发 graphql 服务端项目并且全面负责机构后台前端项目
- 进行项目优化，使 **graphql** 项目能够做到按需加载请求所需字段，按需查询数据库字段，压缩了计算时间，响应大小以及传输时间
- graphql 能够产生大部分前端所需接口，增加了前后端同学的开发效率，减少了后端同学的重复劳动
- 迁移部分老后端接口至 graphql，并且使 graphql 作为其它项目的服务层，以后前端（包括前台，后台，以及机构后台）新接口由 graphql 代替，逐步迁移掉老后端
- 引导前端完成node中间层项目，使前端更加熟悉业务，并且能够更加自由组合自己所需数据
- **使用 graphql 提高前端性能，充分利用缓存，避免冗余 API 查询以及更好地用户体验**
- 完成项目的容器化，并添加 CI/CD，使项目自动部署，减少人力成本
- 优化项目镜像，减少 build 时间，在测试环境的部署时间控制在一分钟以内
- 使用 CI, k8s 结合 nginx，完成项目的多分支开发，即每个分支都有一个地址进行开发测试，完成通过测试合并到主分支，保障项目质量
- 对后端项目使用 typescript 重构，提高项目质量
- **解决高并发问题，对项目添加自动水平扩展，增加了项目的 High Available**
- 使用 redis 完成了 pub/sub，解决了多节点 ws 推送问题
- 添加了 sentry，自动收集生产环境报错，使开发人员能够及时响应问题
- 对日志进行优化，添加 requestId，能够快速在 elk 中定位到相关 log，利于查询性能问题以及定位 bug

学习中心系统

- 2019/04 - 2019 后端 (Node)

技术栈

凡有列出个人必有涉及

前端: react typescript 运维: k8s gitlab-ci elk sentry prometheus 后端: express websocket typescript postgres redis

日常工作

- 维护和优化前后端项目，解决日常反馈与性能问题
- 监控 sentry 系统，及时解决线上问题
- 监控 elk，及时发现慢查询，慢请求，不合理的 SQL (如一条 api 对应 100 条 SQL) 并及时解决问题
- 与产品以及运营开会确定需求

详细

- 解决项目中的缓存崩溃问题，解决项目内存某一时刻的飙升
- 解决项目中的 OOM 问题，当时生产环境，单个节点的内存存在 700M 左右，解决之后变成了 200 M
- 使用 promethes 代替 elk 来统计慢查询，慢请求，并配合 alert manager 进行报警 (另外也可以统计各个项目的 qps，业务线的总 qps 了)