作业：你们的系统使用了哪种服务框架？为什么要这样技术选型？

**优秀作业1：**

1.服务框架当时准备从dubbo和spring cloud两个中选型，最终选定springcloud.

2.选型对比维度：

1）整体解决方案 Dubbo没有自带的配置中心，注册中心基于zookeeper springcloud自 身相对比较完善，有注册中心、配置中心、负载均衡、服务治理 ；

2）底层调用 Dubbo基于rpc，速度快 springcloud基于http，访问相对dubbo慢 ；

3）技术前沿性（趋势） Dubbo已经比较成熟 springcloud技术相对前缘 ；

4）开发语言及开放度 都是Java，开源 ；

5）社区活跃度 Dubbo官方已没有维护，目前一开始重新维护，但基本上能碰到的问题，网上能找到解决方案。 springcloud英文的资料比较少。虽然可能会趟坑，但是业务简单，基本不会遇到什么大问题。

基于自身业务场景，主要只是展示功能，所以速度慢点，影响不大。

优先考虑整体解决方案和技术趋势，最终选择springcloud,虽然可能会趟坑，但基本不会影响业务。真有问题，也是开源，可以查代码解决

**优秀作业2：**

架构的完整度

dubbo的未来定位是专注于RPC领域，成为微服务生态系统的一个重要组件

spring cloud提供了一个微服务的全面解决方案。

1.功能对比： dubbo vs springcloud

服务注册中心 ZK eureka

服务调用方式 RPC http

服务监控 dubbo-admin spring boot admin

断路器 不完善 hystrix

服务网关 无 zuul

分布式配置 无 config

服务跟踪 无 sleuth

消息总线 无 bus

数据流 无 stream

2、 社区活跃度 dubbo

阿里已经重启对dubbo的维护，

spring cloud:提供了一个微服务的全面解决方案，活跃度优于dubbo

3、 学习成本

由于spring cloud都是基于springboot搭建的，其学习成本较低

对于新人来说，能很快的融入团队； 而dubbo的学习成本较大。

总结：因此对于一个业务迭代迅速的团队来说，直接使用市面上的整套spring cloud架构足以可以解决问题；并且spring boot可以跟任何中间件搭配使用，例如ng,rocketmq等，都可以实现业务要求。

至于性能问题，都是通过内网进行Http调用，其效率也可满足公司需要

**优秀作业3：**

前两年我们也对 spring cloud 和 dubbo 进行了技术选型：

1. 从性能效率上来看：

dubbo 使用了 netty 比 spring cloud 的 http 快很多。大概是 2 倍的差距。

2. 从开发效率上来看：

公司同学一般都会 spring，入门 spring cloud 的门槛低且与 spring 融合度更高。dubbo 相对会花一些学习成本。

3. 从社区活跃度来看：

spring cloud 的活跃度明显高于 dubbo。而后续 dubbo 看到了微服务的流行，也逐渐 恢复了更新。

4. 从生态圈上来看：

spring cloud 周边组件很多，通过 starter 自动装配即拿即用。而 dubbo 得自己集成其 他的框架。

综合来看，当年公司选择 spring cloud 的原因是以下几点：

1. 对于性能没有极致的要求。

2. 毕竟是小公司，想尽快产出，减少学习成本，增加开发效率，减少企业成本。

**优秀作业4：**

1.服务框架当时准备从dubbo和srpingcloud两个中选型，最终选定springcloud.

2.选型对比维度：

1）整体解决方案

Dubbo没有自带的配置中心，注册中心基于zookeeper

springcloud自身相对比较完善，有注册中心、配置中心、负载均衡、服务治理

2）底层调用

Dubbo基于rpc，速度快

springcloud基于http，访问相对dubbo慢

3）技术前沿性（趋势）

Dubbo已经比较成熟

springcloud技术相对前缘

4）开发语言及开放度

都是Java，开源

5）社区活跃度

Dubbo官方已没有维护，但基本上能碰到的问题，网上能找到解决方案 ；目前已 重新开始维护；

springcloud英文的，资料比较少。虽然可能会趟坑，但是业务简单，基本不会遇到什么大问题。 基于自身业务场景，主要只是展示功能，所以速度慢点，影响不大。

优先考虑整体解决方案和技术趋势，最终选择springcloud,

虽然可能会趟坑，但基本不会影响业务。真有问题，也是开源，可以查代码解决