Go训练营第一课问题收集

2020 年 11 月 19 日 第一课

注意：大家按照顺序编写，不能在第 1 行编写。而且不能开车

1. 可不可以把 bff 也看做一个服务吗 〈是的，毛老师视频回答了这个问题，可以把 bff 看成是一个按业务聚合基础服务的服务〉
2. bff 和 getway 怎么通讯的 〈grpc〉
3. BFF 和 API Getway 进行衔接这部分的工作是专人负责吗？大概是什么岗位？〈网关通常由运维或基础架构部门维护，bff 通常为具体的业务部门，衔接的话通常是服务发现〉
4. 稿件的 binlog 配合 canal 然后把所有的 sql 发到稿件 job 然后来处理同步到稿件结果的库中，这样理解这个流程没问题吧？(那为什么不直接用 MySQL Master/Slave) 嗯效率高了 还有就是模型变化，pull 变 Push 了，效率更高也更及时
5. 稿件服务利用 binlog 进行读写拆分的例子里，如果只是为了进行读写分离，直接利用 mysql 的主从，查询落到从库做不可以吗？额外引入 canal，kafka 多个组件，必然带来维护上的麻烦，能具体讲是解决了哪些痛点带来的好处值得这么麻烦？〈canal 其实就是实现了主从复制的协议，相当于冒充了一个从节点，然后先接到 kafka异步解偶就完成了复制，之后哪种业务可以根据自己的需要订阅消息，业务根据自己的需要消费这个消息，生成结果表〉
6. 同一个领域后台管理和面向用户怎么拆分？
7. 微服务的数据库数据一致性怎么保证 ---MQ 本地消息表确保 --或者参考 seata -> 本地事务+消息队列〈数据一致就两个方法：1 强一致，2 最终一致，看业务场景选，强一致方法就是事务，事务又有分布式事务：两阶段提交（2pc）等。微服务中经常保证一个方法的原子性，服务之间靠补偿服务实现最终一致〉
8. API Gate 如何解决客户端多版本收点的问题？明白。----这个问题可以用多租户的灰度发布，根据租户代码不同，应用版本不同来实现网关流量的分发，其实就是流量染色（这个毛老师的多租户估计就是讲这个）那不同 API 版本的 BFF 是独立的吗?

   说的是客户端版本吗？客服端？ 一般都是服务端会向下兼容，或者客户端到一定低版本后，会自动退出，强制用户升级，另外多版本问题，可以放在 bff 层来做，会更合适一些。BFF 本来就会对业务做纵向拆分，b 站听毛老师说是分成了三个我记得。
9. 如何解决服务间调用的闭环问题 是不是服务间的调用没有讲? 理论上是 DAG？总得有一层是乱的

   闭环一般都是通用服务下沉。 服务间的调用跟业务相关，一般都是 rpc 调用。DAG 框架我没用过，同事说很难用，估计封装的不好吧。。。
10. 网关只负责基础的身份认证？是否包括 API 的访问权限控制或数据的访问权限控制？网关和 BFF 怎么划分？【前置判断】

gw 负责的是下游服务，统一收敛的公共能力，比如你提到的鉴权，防水墙(安全问题),缓存，频控等等。公共能力如果每次变更，下游所有服务都需要在发布一般，所以才有了 gw 这一层。 并且这一层可以各个接口的监控，大盘的整体健康度等。所以是很有必要的。BFF 你可以没有很好的理解，一般公司可以不会有这一层。 因为 b 站业务终端场景复杂，tv、mac、iphone、aphone、web 等等。为了适配多端场景，根据不同业务抽象出来这一层。放在后台做，目的就是一些简单的变更等，都不需要终端发版等。也方便下游不需要为了适配，而编写不同的代码。

11. 在网关中单节点的限流或者是熔断方便实现， 如何实现分布式限流？分布式熔断控制？🎉内部服务的调用呢N+1

两个是不同概念，毛老师后面会讲。 内部服务调用，你想问的是会扩散的问题吗。

12. 网关挂了咋办 +111111，网关有状态吗?

没有状态，服务之间一般都是没有状态的，一般都是存储才会涉及到有无状态。 网关一般不会挂的，1 代码也有各个单测，测试用例等通过一些手段来保证代码等质量

2 部署发布阶段会有灰度等策略。并且服务机房间就是本来就是集群。如果因为过载等会有监控，限流，熔断等手段来保证，而且一般都是下游业务相关，网关不容易挂，再有如果机房及时关了，一般网关都会有跨机房容灾的。

13. 在 BFF 中是否是只做针对查询的聚合，不会涉及将查询、创建与更新等混合操作的聚合，如果会的话，如何保证事务一致性?

可以看下 9 回答，是的，bff 一般都是数据整合。 事务一致性，毛老师应该也会说。

14. Q:为什么要将用户 token 转成 jwt？没有转换 （jwt 可以携带信息，user_id+权限）

因为下游服务可能会需要用到这些信息，方便下游。具体得看业务场景，也不一定要这么设计。

15. A: 个人觉得 对外的做用户 jwt 认证完 就不用在去做认证，省去内部 rpc 的验证 [jwt token 通常不用作用户 token 的原因有两点，1 如果加密用的 secret 泄露，将极容易被攻击；2jwt token 中携带的明文信息会暴露部分系统设计信息，不利于系统安全；]
16. 用户 Token 换成 JWT Token 是什么意思呢 [用户 token 是用户身份认证系统签发的用于确认用户身份的东西，只有在 API Gateway 和户身份认证系统会用到，在其它业务系统用不到，因此 API Gateway 会在将请求转发给 BFF 层之前，将用户 Token 换成下游业务系统需要的 jwt token。]
17. 我理解在 API Gateway 做鉴权的时候，不是应该根据 client 传递过来的 Token 解析得到用户信息吗（如 uid、name 啥的）？ 还是下沉到 BFF 层去做？为什么呢 [这里一定要注意身份认证和鉴权是两件事，身份认证解决"你是谁"的问题， 鉴权是解决"你能干什么"的问题。通常来说，身份认证是由 API Gateway 来做的，鉴权信息可能会在 API Gateway 中添加到 jwt token 中， 具体的权限判断逻辑在下游的业务系统中。]
18. 发送时要保守，接收时要开放 这个不是很懂？需要做分包发送吗？["发送时要保守，接收时要开放"是指系统和外部系统做交互时的一个指导性原则，和具体技术无关。这里可以理解为，对外部系统应该尽可能的宽容(开放)，对内应当尽可能的严苛(保守)，是从人类社会交往的"严于律己宽于律人"原则类比而来。]
19. ssr 服务端渲染怎么理解 [服务端获取数据后直接渲染 html 文档，前端不再需要调用接口]
20. 如何解决多版本 多网络问题 根据不同的网络给出不同的判断

21. 大前端 - 网关接入 - 业务服务 - 平台服务 - 基础设施，这里的平台服务具体是指什么?

--A: 基础服务，如消息队列服务，缓存服务等功能服务。

22. 毛老师，请问你关注、使用过 GraphQL 吗？看到你说的 CQRS，Compose 接口什么的，觉得这就是 GraphQL 的应用场景呢。.

-- GraphQL 和 CQRS 没有关系,compose 是 GraphQL 的应用场景

23. gprc 怎么上公网?

--grpc 提供对外的端口提供服务即可，注意安全鉴权这块

grpc的"传输介质"是http2, 只要通信的两端支持http2协议，就可以使用grpc，和内网，公网没啥区别。

24. 老师，这个有两个问题，BFF 这块：1、对 provider 提供过来的实体字段要进行一些裁剪和组装（组装来自两个服务的数据实体，然后一起返回到前端 app，如果其中有一个异常了，怎么处理），有什么推荐的做法呢，特别是裁剪，能细一点的工程实践吗，不用在单独定义实体吧？2、比如一个评论列表，要调两个服务，一个是评论，一个是用户服务，要通过评论服务里面的用户 id，组装出 用户名，和用户头像，用户的等级，这个你说的批量去捞，具体有什么好的实践做法呢，根据这个列表是 10 个 用户 ID，去账号服务里面 去 sql in 查询吗，或者其他的更好的做法?

-- 第一个问题，数据组装这块要按照具体情况来定，如果涉及到核心业务直接抛出异常错误码，如果不是核心业务返回空数据，前端兼容数据展示。数据裁剪得看客户端或者前端的数据格式来定，因为业务不一样还是定义实体来区分比较好。第二个问题，可以考虑并行方式用 id 去调用后端原子服务。

25. 微服务兼容性问题，发送时保守，接收时开放，可否再举例详细说明下?

-- 这样理解对自己严格对他人宽容，自己在发送数据时按照既定的格式或者规范进行格式化，而在接收方时为了保证自身服务的健壮性，在解析这些数据时兼容更多的版本格式或者可能的数据格式，涉及到多余的字段就可以忽略

26. 后端 service 务不直接对外提供 API，每次都通过业务 BFF 对外提供服务，那么 service 新增一个接口，BFF 也需要开发一个接口去调用，如果这个接口比较简单，那么开发工作量是否比较大?

-- BFF 上写调用接口的代码就是搬砖的工作，BFF也支持裸透传，哈哈哈

27. BFF 之间会互相依赖调用吗?

a. BFF 之间不会进行依赖，会按业务域划分，并且统一调用 Service；

28. 网关层的最终版架构是不是又是单点了?

a. 网关主要逻辑非常少，也可以按业务域划分集群，当然也可以通过到 Sidecar 方式，需要看公司或者业务场景定位再决定做成什么样的网关；

29. 服务众多，测试环境怎么搞，怎么联调?

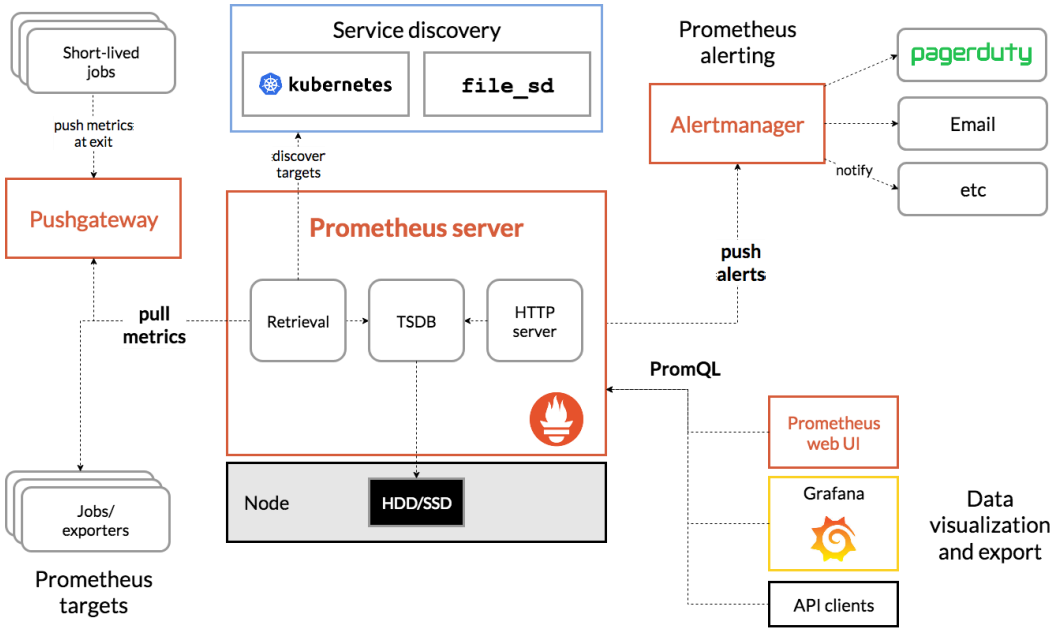
a. 这个需要看基础实施的完善程度，比如通过 CICD 自动化构建部署，或者 jenkins 手动构建部署，分支多时可以通过服务流量染色解决联调；

30. 服务拆分后，某些需求尤其是大权限的，发现因为数据需要，服务间互相调用数据量非常大，聚合起来特别繁琐，这个有什么好的解决办法吗?
    a. 这个需要首先需要把 DTO 设计好，通过 Batch 方式请求，缓存+DB
31. bff 能用 nodejs+GraphQL 来实现不？
    a. 当然可以，不过 GraphQL 在 web 方式使用非常方便（字段按需），客户端一般定义固定的 Model 类，需要做好的 scheme 生成工具；
32. 修改大表结构后，通过 binlog 前端数据同步数据缓慢，这个有什么好的解决方案吗?
    a. 加快刷盘，拉日志不慢，在从库回放慢，可以临时修改从库刷盘参数（牺牲安全性），加速导入；
33. token 到 jwt 到 userid 的过程说一下，jwt 除了放 userid 还放啥信息?
    a. 鉴权通常分为：User、OpenAPI、LDAP 这种，除了 uid 可能还有类型、用户名、角色等等
34. 注册中心 consul 去 reload nginx, 需要改 nginx 配置文件吗?

如果从consul拉取的服务端的ip列表和之前的不一样，是一定要修改nginx配置文件，然后reload的(nginx -s reload)

35. promethues 架构是怎么样的?

官方的整体应用架构图如下：



若有其他细节疑问可私聊我沟通。

36. docker 审计方案怎么做?

问的同学是运维安全向的吗，是指的容器内操作的审计，还是容器镜像拉取推送呢。

大体整体如下：

（1）容器内操作的审计：可以走传统的堡垒机方案，逻辑是在连接容器时先在堡垒机运行录制程序或脚本，然后对命令进行记录或筛选。

（2）容器镜像拉取推送：需要实现代理，在仓库和用户间使用代理对请求进行过滤，案例的话可以具体实现可以参照harbor的代理模块。

37. 为什么需要 BFF 呢?

服务化后后端服务基本切割的比较原子级，大多无法直接应对业务端的访问。而 BFF 就承担了业务聚合层的功能，需要针对不同的端进行业务数据裁剪、聚合和再组。

38. 为什么需要把多个请求打包，分开发送给前端有什么问题吗?

需要上下文，单纯看问题没法确定，麻烦私聊我沟通。

39. 有没有微服务实战的项目去加深理解。

可以看看 https://github.com/bilibili/kratos，自己多搭建几个服务，实际跑一下。如果对实践项目有疑问，可以看看我出的 《Go 语言编程之旅》的第三章有相关的实践内容。

40. 如果用 go 做 web 服务,那么还需要 nginx 转发吗?还是说直接使用 go 接收请求并处理.

已私聊回答你。

一般都需要。go在高并发方面有优势，而nginx在性能，安全，缓存，静态文件处理，稳定性等方面都是go比不过的。并不是用go一定写出来，而且市面上已经有nginx把这一部分做的非常好了，其他人就没有十足的动力重复造轮子了。

41. proto 声明文件如何在 服务间共享

一般有专门的一个 git repo 进行管理，进一步升级，可以集成在服务接入平台上来进行管理。

42. 每个微服务都用不同的 DB，那这样会有很多的 DB，db 之间的数据怎么去共享呢?

db 之间不应该互相共享数据，尽量用 id 来进行互相关联。部分互联网应用采用反范式设计，会做一些冗余，例如淘宝的订单中的地址信息等。

43. gRPC 可插拔能稍微解释下吗?

可插拔是一种架构上的概念，如果你的功能是依赖于接口而不是具体实现，那么你就可以在不修改业务逻辑的情况下更换底层任意实现。

44. grpc 没有注册中心的话，怎么去做负载均衡呢

- grpc 的 runtime_notifyListNotifyAll
- runtime_notifyListNotifyAll

45. 接口 batch 调用怎么做

例如获取用户信息，在 request 中可以传 user_id_list：[1,23,323, 2121]，server 端同时返回多条用户信息。db 查询可以用 where user_id in () 来实现

46. 微服务多层调用后公共返回值和业务返回值怎么设计，是不是需要所有服务约定共同的公共返回值，遇到可透明代理接口时，可以直接用下游服务返回值对外服务，还有追踪参数在这种情况下怎么处理（ps:如果追加追踪参数就会有 decode 和 encode 过程）

可以的话尽量避免多层调用，因为这样容易导致级联失败，迫不得已的情况下，可以要求你的对接方提供一定程度的数据拼装服务。公共返回值这个在说啥没看懂，是说统一的错误码么?
trace 信息在现在的 rpc 中一般放在保留字段里，例如 thrift 的第 0 个字段。

47. 微服务拆分服务与 DB 之后，服务与服务之间的调用怎么保证数据的原子性呢

----强一致用分布式事务，不需要强一致可以用 CQRS 之类的方案实现最终一致

48. 说了这么多概念，我们怎么自己加强基础，然后在项目中逐渐落地?

---- 这就需要实践了，比如如果有个好机会去独立设计一个项目的架构，那么就可以思考一些技术能不能引进在架构中，调研的过程也是学习的过程。

49. 客户端不升级，导致多版本， 业务层接了网关之后， 问题不是仍旧存在吗？业务层仍旧要提供多版本服务

----可以通过 BFF 层来做客户端区分，业务层可以按自己的方式提供数据接口给到 BFF，BFF 层根据不同客户端的需要进行数据处理，然后根据不同的客户端输出它们的数据。

50. 安全这块，服务接口之间的调用，如果连接已经使用 tls，还需要使用签名进行参数完整性校验吗？ 防止篡改!

a. 见这个解答： https://www.zhihu.com/question/52392988

51. 微服务需要做数据库主从吗

---- 如果每个服务要使用自己的独立的数据库的话也要考虑大数据量，数据冷热分离的问题，这个和实际业务场景相关。

52. 哪些要 BFF 哪些服务要放到后端，能不能举例说明一下，不会存在转换效率问题吗

--- 比如前端需要各种门店的数据，但是每个门店在 web 端和 mobile 端的展示方式略有不同，假设 web 端不需要门店标签信息，而 mobile 端需要门店标签信息，这个就可以放在 BFF 中进行区分，而后端 API 不需要区分请求是谁发给自己的。

53. 二次提交解决一致性问题，2 个服务好搞，自己开事务，调对面，然后确认一起 commit，多个服务怎么弄，略复杂

---按照 2PC 的定义，多个服务就是需要多个服务作为参与者，在协调者的协调下同时 commit 或者 fallback，这个自己实现肯定是比较复杂的，可以考虑一些开源的组件。

54. 分布式事务怎么做?

解：tcc，seata

55. grpc 客户端流式，服务端流式，双向流式，在应用中的具体实例

解：

- 客户端流式：请求里 req 包比较大，但 response 包比较小
- 服务端流式：请求里 req 包比较小，但 response 包比较大，可以用于定时任务拉取分页信息，缓存中拉取全量数据等。包太大一次返回对服务压力很大，所以在不要求对时间情况下，可以通过流式返回
- 双向流式：比如 im 通信，watch 事件

56. 服务的版本是怎么划分做到多个版本共存和兼容的。url 里面加版本么 bu

- 参考 gitlab，划分 v1，v2 的 url。/api/v1/user, /api/v2/user

57. 老师 CICD 的时候，为啥不用 jenkins 呢

- gitlab 比较方便

58. 如果没有移动端，只用 web，是不是大前端就做 BFF 了?

- bff 只是一个分层聚合后端服务接口，这个分层可以大前端做，也可以后端做

59. GRPC 如何推进？我们之前做过一次 GRPC 推动但是最后还是无疾而终了，现有 HTTP 调用方式公司内部工具链比较成熟了，例如网关、服务暴露等等但是 grpc 没有相关服务支撑，改动会比较大，切换的时候就推不动了。

柠檬：组件的**业务价值是首要性的**，如果现有技术体系很好的满足了业务要求，且开发效率也很高，**没必要非要切换**。一般来说由下到上的推进是比较难的，从上往下的推进会简单很多。

不管如何推进，要明白为什么要替换，有什么收益，要熟悉其优点/缺点/坑点，熟悉其性能，适用场景。做业务适配，效率工具，整理教程等。

先从小的点做起， 新技术虽然炫酷，但也包含着不稳定性以及不好把握，不上线的话，不知道有什么问题。从新服务，非面向用户的后台服务，非核心服务逐渐推进，出问题的话，也不会影响线上，有足够的时间排查积累。

60. 压测性能会讲吗

目前看课程安排没有

61. CQRS 数据同步链路看起来很长，数据在读的时候，状态会有问题吗？（稿件修改和稿件结构那个场景，稿件修改要经过 binlog - canal - kafka - archieve db 等多个环节

柠檬: 这种架构看应用场景。**状态会存在短暂不一致，但是在可允许范围**。

1.稿件状态有限，且变更不会很频繁。

2.实际应用中给别人看的信息，不一致时间窗口小于 2s，基本不会感知到。如果是用户自己看，可以读原数据。

3.当然实际场景会比 ppt 中描述的复杂，除了这一套架构外， 还需要考虑丢消息的问题，是否需要进行一致性校验，全同步等。

62. BFF 有什么推荐框架？GraphQL 用的多吗?

柠檬强答：这块我了解不多，我们是自己封装的业务接入层。

graphql 据我了解大公司后端用的不多。

63. 服务层互相调用可以么？ 怎么阻止服务层调用 BFF 层 和 BFF 层调用 BFF 层导致架构混乱

柠檬：服务之间是可以相互调用的，非要乱调的话，也没法硬性阻止，调用需要遵守一定的原则，整洁架构之道这本书有说明。

无依赖环原则：调用关系不应该成环。

稳定依赖原则：依赖必须指向更稳定的方向。

稳定抽象原则：一个组件的抽象化程度应该与其稳定性保持一致。

这些是服务之间调用时需要考虑的原则。比如支付中心就不应该调用业务活动，基础数据就不应该调用首页服务等，越是基础的服务，其依赖应该越少。

架构需要经常进行梳理，排查不合理的地方。比如人工梳理调用图，梳理 trace 图。

64. 使用 mq 进行服务调用与使用 rpc 进行服务间调用都适合什么场景?它们有哪些相似与不同点?

柠檬: mq 不能叫服务调用，是一种解耦合，异步的方式，服务发出 mq 信息后，就不再关注了，接收者可以有很多。rpc 是强依赖，同步的方式。

大部分业务逻辑直接用调用 rpc，比如我先调 A， 再调 B, 调 C 等。

mq 一般用在异步场景，解耦， 削峰等。

1.发出领域事件：比如注册，登录，充值，点赞，评论等关键事件。对这些事件感兴趣的服务可以自己去接收处理，处理不处理，处理是否快慢，都不会影响核心服务。

2.异步削峰：可以看作是同步 rpc 的一种降级方案，用于匹配快慢系统。比如活动领券，一般业务服务性能可以很高，但是发券服务要操作 db，有很多校验，甚至需要匹配第三方厂商，一般性能会比较低。高峰时直接发券会打垮发券服务，所以先告诉用户请求成功，发券会稍后到账，接收消息进行匀速的发券处理。

3.消息补偿：同步调 RPC 可能会网络失败，且不可能一直重试，可以将事务的下半部或失败补偿放入消息队列，由另一个服务去保证一致性，参见分布式一致性中的本地消息表和事务消息。

65. 微服务之间调用，也是走网关吗

柠檬：不会走面向外网的那种网关，这个问题需要分情况。

网关的定义是什么：不同网络之间的转换的设备。一般网关用于不同系统的边界，做协议转换 / 统一入口 / 安全校验等。

下半节会讲服务发现 / 负载均衡的模式，有客户端发现，服务之间之间调用，服务端发现，通过中间层服务进行调用，比如内部服务统一通过 url 调用 nginx，nginx 做服务发现，负载等。

但是大部分情况都是用的客户端发现，服务之间 ip:port 直接调用，这样可以减少中心单点问题。

还有种情况如果公司比较大，有不同的大团队，可能你是业务服务需要调用别的团队的支付服务，那么就需要走他们的网关了。

66. 业务限流为什么不在 LB 层做，而要下沉到 API 层。

解：LB 层往往是业务无关的，即 LB 层不关注具体的业务场景。而不同的业务限流，会有复杂的规则，例如说依据 CPU 情况，网络负载等进行限流。如果要在 LB 层限流，即意味着 LB 层需要掌握这种信息，这对于 LB 来说负载过于沉重。根源还在于说 LB 这一个环节所具备的信息太过于优先，尤其是四层 LB，基本上就没有什么业务信息了，根本做不来。

但其实，硬要在这层做，并非不可以，只不过 LB 无法做复杂的限流。例如简单的，针对 API 硬标准就是不超过 100/s，那么的确可以在七层 LB 的时候，考虑支持的。只不过收益不高而且又导致 LB 不再是纯粹的 LB，影响 LB 本身的性能。通常 LB 是全局性的，意味着你针对某个 API 的限流的逻辑的开销，会影响到你全局的 LB 的性能。

67. 编排是什么意思？组装的意思吗?

解：容器编排其实是指自动化容器的部署、管理、扩展。举个例子，使用 k8s 编排容器，那么当某个容器突然死掉，那么 k8s 就会自动启动一个。k8s 是一种声明式的 API，即我们声明我们期望的集群是一个怎样的集群，例如有多少个实例。那么 k8s 就会调度管理容器，达到我们所期望的状态。

68. 微服务之间有各自的资料库，当有需求需要跨服务的 query/join 时该怎么办呢？让前面的服务客户端自己处理不会有资料传出大/效率低的缺点吗?

解：这个问题可谓是击中了微服务一个老大难的问题。我可以更加深入描述一下这个问题，除了题目本身提到的跨服务 query/join 的问题，还有一个更加可怕的问题，即一条链路上重复调用某个服务多次。例如 A 服务调用 B 服务，而 A 和 B 服务都调用 C 服务，因此在这个过程就会调用两次 C 服务。

从一般的设计上来说，我们会使用聚合服务来完成跨服务的 query 和 join。BFF 也具备同样的性质。

就是我们在专门的领域服务至上构建一个聚合服务，聚合服务来调用下层的领域服务。领域服务来完成数据的组装，而后拼接返回给客户端。

这里面涉及到一个灰色地带的问题。即这种聚合数据的逻辑，我们是否认为是客户端应该 own 的逻辑。如果是，那么放在客户端也是合理的。

无论是聚合服务处理还是客户端处理，都难以避免响应大的问题。但是，就我们实践经验来看，造成性能瓶颈的往往不在 RPC 请求响应的传输过程，所以，这些缺点是可以忍受的。

而从优化的角度来说，我们宁愿提高服务端生成响应的速度，也不愿意为了性能而将业务逻辑放在不合适的地方。

69. http -> rpc 一般在哪做转换➕1

解：api 网关，即 api-gateway。

70. 用了 k8s 后，还需要用 gateway 吗？直接用 k8s 的 service ,ingress 可以替代 gateway 吗。

解：并不能。目前来看，gateway 的很多功能，都是 k8s 的 servic，Ingress 所不支持。比如说熔断，协议转换之类的。

71. 链路长了 测试怎么解决？ 全链路压测怎么做?

解：目前长链路的测试还是比较复杂的。核心在于，要有一个东西，能够把链路整个串起来。比如说，通常的 tracing 工具，我们会使用 traceId 来追踪。又或者，在链路开端生成一个代表本次请求的唯一 ID，而后使用链路的这个 ID 来追踪整个链路的执行情况。

而还有一个难点在于，长链路的话，需要有一个稳定的测试环境。通常而言我们修改或者开发新功能，不会是整个链路都修改，而是中间某几个环节变更，因此，在测试的时候首先要保证，没有变更的那些环节，是稳定的，能够如预期工作。而后测试的工作就不会因为那些环节出问题卡住。

而后，如果可以的话，应该考虑使用 mock 服务提前进行测试，而不是一直等到全链路都准备好才开始测试。

至于全链路压测，难点在于环境搭建。我们先考虑最简单的读链路压测。读链路压测，因为不涉及数据变更，所以环境可以直接通过生产环境横向扩展完成搭建（加服务器，加服务器！加读库加读库！）。而后，我们需要在请求里面带上一个标记这是压测请求的标志位，在整个链路中间传递。这些请求应该只会被用于支持链路压测的服务器所处理（避免影响正常请求）。

而写请求的链路压测，则复杂得多。难点在于准备数据库。一般而言是使用影子表或者影子库。影子表使用比较简单，但是因为它和正常的表共享一个库，所以天然就会影响正常请求。影子库要好一些，可以规避对正常请求的影响。无论是影子表还是影子库，绕不开的一个问题就是，真实生产环境的数据是极多的，准备这种影子环境，只能考虑将一部分数据导入过来。于是就涉及数据间的一致性。例如你准备用户表，就要将相应的用户的订单数据也弄出来……

此外就是，压测的另外一个难点是构造请求。通常做法是录制流量而后重放。即录制真实用户的流量，而后重放这些请求。这方面有一些合规性的问题，即有些数据，在某些国家的隐私标准之下，是不允许这样重放的。

72. grpc 或者 protobuf 同一接口多协议版本如何兼容性；

解：单个 TCP 端口上提供 HTTP/1.1 和 gRPC 接口服务，可以在接口检测 Content-Type 是否是 grpc，检测是否是 http/2，根据不同的协议转发到不同的服务处理

73. 一个服务的数据库通常是拆分成什么样的结构，在 k8s 是用是 statefulset 部署在内网的还是单独部署在一个公网的环境上的，有做分库分表或者主从之类的么

解：数据库的拆分要根据业务来拆一般来说不要跨业务访问库（通过服务接口），分库分表取决于业务要解决的问题，比如单表数据过多造成单表读瓶颈考虑水平分表，单表有大字段考虑垂直分表，单实例写瓶颈考虑分库分实例

74. b 站现在目前是怎么实现动态加载 nginx 负载均衡 ip 队列的？现在还是视频里讲的那种方式么？

75. 去中心化，强调服务独立性，那么那些微服务间互相调用约定好的数据结构怎么规范，有个类似 common 库那种的么，还是别的解决方案？出现需要多表联查，但表在不同微服务的数据库里，数据库的性能是不是会比 rpc 性能更好？

解：查询数据库尽量保持简单，不要为了微服务而微服务去拆分库实例给业务造成不必要的操作

76. 可以讲讲 serverless 么？跟微服务有什么不同？

解：serverless 是一个比较大的范畴，FAAS/BAAS 都是 serverless 的一种实现，微服务是一种架构主要本质有两点：解藕，分布式

77. 现在很多微服务框架，怎么选，APIGETWAY 里的功能比如限流等都是现成的吗？只要自己写业务代码？

解：有的，具体怎么限流要根据业务场景，一些比较特殊的业务场景需要自己增加限流逻辑

78. 微服务拆分完了之后，假设有一个聚合的数据展示列表的业务场景，需要跨多个服务去做查询，那么分页，还有跨服务的查询条件怎么去处理呢？

1.以外部服务条件为基准，通过本服务 id 排序查并询数据，查询完之后，通过外本服务键加上本服务条件需要获取本服务数据再进行组装。2 为什么要本服务 id 进行排序呢？主要是因为怕本服务分页时候出问题。

.缓存外部需要查询的数据到本地，但需要考虑数据一致性问题。

79. 同一个微服务如果起了多个节点，那他的上层都需要做负载均衡来决定去 call 哪个服务吗？比如多个同样内网微服务节点靠 bff 层做负载均衡，多个同样 bff 节点靠网关层做负载均衡？

如果进入 bff 节点的流量，经过了网关，那么网关可以进行负载均衡，但首先它要支持这个能力。如果 bff 节点到微服务没有经过网关，那负载均衡就应该是代理来充当负载均衡的角色。

80. gateway 鉴权是直接调用账号服务吗？和登录注册服务的交互是怎么样的呢？

gateway 应该自有一套账号体系，独立于账号服务。因为 gateway 使用账号的主体是服务，而账号服务使用账号主体是人。那就应该就不存在与登陆注册服务交互。

81. 能简单讲解下横向切面的概念吗

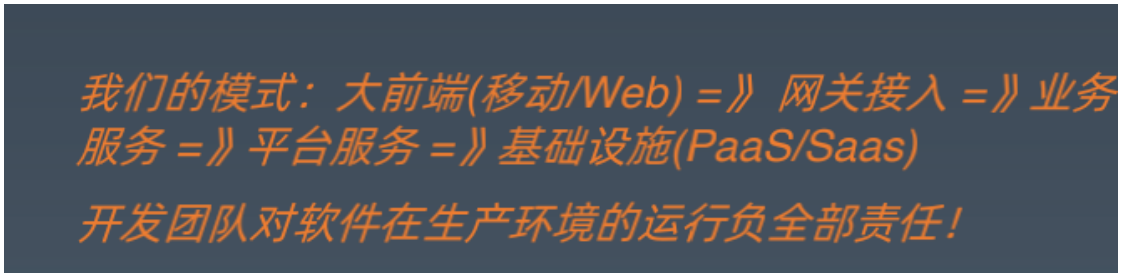
82. 是 java 里面 spring AOP 的意思么？如果是的话，你可以这么理解：如果一个类相当于一个柱体，在方法的开始切一刀，就是一个切面，方法结束也是同理。其实就相当于 Filter，但是简化了 Filter 的写法，通过更方便的形式实现，如 java 里的注解。 微服务间通过本地事务及消息队列来确保数据一致性，那消息队列的稳定性、健壮性如何保证?

从生产者方面，失败重试。从中间件方面，1.进程监控，2.队列监控，包括消息堆积、消费者存活数量等，从消费者方面，消息幂等。

83. API GateWay 的 enovy 是代替 nginx 的吗?

是的

84. 微服务可不可以理解成下图中的平台服务，BFF 理解成下图中的业务服务对么？再者，BFF 一般以什么形式导出的呢？RPC,还是 Http。



这种模式比较类似现在说的中台模式，平台服务是有业务属性的能被多条业务线公用的服务。BFF 是纯数据拼装服务，里面没有太重的业务逻辑。业务服务是各个业务线自己的重业务逻辑。BFF 的出口协议可以随意(你可以说 gRPC 性能好，也可以说 http1.1 更直观好调试)的，按毛老师昨晚讲的在 BFF 前面还套了一个 envoy，怎么方便怎么来。

85. Api GateWay 到数据库，整体流程中哪些部分使用 go 比较有优势，为什么?

目前 Go 还是有 GC 和调度，在一些延迟敏感的场景不好用，比如入口网关，或者 dbproxy。个人认为 Go 最擅长的还是在业务模块。或那些对延迟不敏感的基础设施。

86. 老师昨天说不要再一个项目下有 utils/common 之类的，容易变成垃圾箱，那比如说转换时间格式、个性化的整合信息，只有这个项目独享但是被多个地方调用了咋办呢。做成 package 的成本也是一样的吧，如果 package 多了还不容易区分

可以参考 github 上一些国外公司的做法，比如 dropbox 的 util 库等等。

87. 治理去中心化什么意思，没有太理解，能举个例子么

88. grpc 怎么配置多个双向认证的证书来进行细粒度的权限认证

89. go 怎么通过 grpc 实现异步调用

90. (1)为什么移动端会连不上 http2.0（2）grpc 不是不能直接通过 http 请求访问的吗，要通过类似 grpc-gateway 这种转换才行，那怎么在不修改代码的情况下提供公网访问？（3）grpc 怎么分别提供 http1.1 和 http2.0 的服务

97.最大限度的容忍冗余数据希望举个例子

98. 毛老师您好，我的问题是：将巨石应用按业务拆分成微服务之后，各个微服务之间有一些共通的功能怎么来实现呢，比如拆成了订单和用户两个微服务，提供给后台管理的 CRUD 功能，是这俩微服务都独立实现一次吗?

99. api 自动化测试提到使用 yapi，我们最近也想做自动化测试，还有相关这类好的工具推荐吗？我下来对比调研下，另外 b 站目前就是实际使用 yapi 来做的吗?

100. (1). 单体到微服务过渡阶段, 是部分模块抽取之后, 原来的接口服务调用微服务, 还是, 整个接口里的服务抽完, 一次性处理 (2). grpc 上公网是指移动端和服务端通过 grpc 通信吗,这中间有什么坑没有

101. tcp 链接 a 服务发送了 A1 消息给 b 服务 b 发送了 B1 消息给 a 服务 期待收到 B2 消息 此时收到了 A1 消息这么处理

102. 在讲 API Gateway 3.0 的时候说到由于代码可能会有 panic 的情况，想到如下问题：

1）如果真的由于这种 panic 而导致的进程退出，该如何排查最终导致 panic 的问题所在呢，毕竟当时引发此 panic 的进程已经退出了，这里可否提供一个简单的 case 呢，真实实战一把排查定位问题的过程

2）平时写代码时，该如何避免这种由于 panic 而导致的进程退出呢

3）有没有这种情况：如果某个场景下真的出现了不知道的异常错误（对于异常处理的代码来说相当于是未知的错误），但也确实成功地捕获到了它，这时是该让进程退出呢？还是忽略掉未知错误呢?

103.毛老师，你好，闭环去做微服务，这个闭环该怎么理解？每个单独的服务独享数据库，那每个服务的数据库，之间的关联关系呢?

104。服务之间通信，应该选 http 还是 rpc，gRPC，什么场景使用 http，什么场景使用 rpc，他们各自的优缺点是什么

跨机器的通信，需要用到网络。在7层网络协议中，一般使用TCP，UDP和TCP之上的http协议。（暂不引入其他的协议）
RCP（远程过程调用）是一种逻辑上的概念，需要借助网络来通信。RCP更加关注调用者和服务提供者，而有意忽略通信协议和消息格式。（而且通信协议和消息格式一般都是可插拔/替换的）
gRCP就是这种思路的一种具体实现。采用的通信协议为HTTP2，消息格式为protocol buffer。
与protocol buffer同级的有json，xml，msgpack，核心就是如何表示消息。
与gRCP同级的有thrift,主要关注消息的发送，接受和传输，依赖TCP或者HTTP。
http无法与RPC直接比较，因为两者是不同的概念。http只负责把消息送到，而RCP更加抽象，涵盖更广。

105. 在说到 Mircoservice 划分时提到了 CQRS，且用 binlog 订阅的方式同步数据到指定的库中供业务服务查询，想了解的问题是：

1）如果业务服务在后面某个版本中需要查询更多的字段信息，而这些信息之前并没有同步过来，此时该如何解决呢？目前能想到的是写个脚本去全量同步过来，还会有更好的解决方案吗

2）如果写脚本同步，这个脚本又该写在哪里呢？提供查询的业务服务中?

3）还有就是读取数据的方式问题，应该不可以直接连对方的 DB 吧，还是让对方提供接口的方式呢？感觉都不太合理.

106. JWT鉴权这一步一般是由GETWAY来做，颁发JWT一般是在用户登陆的时候，由APP(用户集群)来做。而生成JWT和对JWT验证是需要秘钥的，这是否意味着GETWAY和下游的APP(用户集群)拥有相同的秘钥?