

Go进阶训练营第二课问题收集

Go进阶训练营第一课问题收集

文章书籍记录

1. 最近团队在做微服务框架选型，主要在比较go-kit, dubbo-go, go-micro这几个，业务场景是物联网平台和运维平台，可能会进行私有化部署，想请教毛老师对微服务框架选择有什么建议。<首先需要确定是否有历史包袱，如果有java包袱，那么dubbo-go较为合适，若是新项目可以考虑go-kit, go-micro目前版本变动比较大不太建议，另外可以关注一下go-zero,kratos,go-chassis, 另物联网通常需要mqtt, paho系列的客户端非常不错>
2. BFF层定义的API多种多样，比如一个根据订单id查询、一个根据用户&时间等多种多样的维度。SERVICE 层如何兼容各个BFF API? <接口单一原则，尽量保证一个接口只做一件事，但你的例子：订单id, 用户, 时间 这一看就是订单服务，所以肯定是一个订单service（至少是订单数据查询维度的服务就基本可以了，不要再拆了）>
3. 关于 gRPC 的 proto 文件维护的最佳实践是怎么样？比如说 rpc 的调用方和被调用方都要在代码库维护一份 proto 文件的代码吗<统一repo维护，可通过一些ci工具触发对应项目hook进行具体语言的代码生成>
4. 目前是做java开发，想转到go，但是感觉看go的文档比较不习惯，有没有什么快速入门能够先从简单项目上手的方法？刚参加工作不就，很多分布式方面的知识都不懂，对于后面要讲到的内容想做一下预习，建议看一些什么？<熟悉一门语言，首先就是了解其语法特征，而官方文档是最准确的资料。其次第一课推荐过一些基础学习的知识用例，可以完整了解整个语法特性，然后就是多看一些优秀的项目了解别人实现的方法以及流行的风格，第一课讲了从单体到分布式的演变过程>
5. 学习的总结是推荐放在哪里或者什么工具归档<onenote、有道笔记、语雀、git都可以，一定要有总结沉淀出自己的内容>
6. proto buffer 改动后如何做到多端共享
 - a. 据我现在在做的事情而言，直接把pb文件给其他端就行
7. gRPC的强约束是指什么，有什么作用
语言约束，都通过pb定义接口。字段变更，下游使用的服务都需要更新。作用：对比restful
8. health check 一般多少的频率，ping啥频率
ping就是health的一种方式吧。一般是分钟级别，加上可容忍的心跳数，再加上毛老师说的子集算法
9. 外挂方式和主动
10. 方式有什么优劣区别吗
外挂方式就是容器内部服务器起来，就已经往注册中心注册好服务了。代码主动注册。需要多写代码吧
10. 外挂方式如果是感知到新添加进来的节点的？
外挂方式会订阅注册中心的信息(自己相关的信息)，当有新节点添加，会往注册中心注册，就会感知到了。
11. 服务端发现 LB 可以部署多实例来分摊负载吗？直连的话，重新发布时总有一段时间有客户端流量打过来
可以分多实例。一般服务发布的时候，可以调整权重来发布，或者用热加载。

12. 用eruka不用etcd作服务发现是因为cp性能没有ap好吗？

不是因为性能问题，服务发现ap更重要些，对于一致性来说，可以用health check来相对保证c。所以选项用了他

13. eruka只用做服务发现吗？牺牲了一致性那么相对consul，etcd是不是少了作为kv存储的功能？

术业有专攻，可能你了解的etcd会拿来做一些存储，但也是轻量级的。

14. healthcheck 是每台机器 ping 他依赖的所有服务的ip列表么？

14.1 先讲为什么要ping接口。

例如http://xxx.com/ping，探测路径/ping是一个约定，生产服务必须有探活接口。

理由有2个，1保证应用级别存活结果可知，2响应超时多是资源（cpu/mem）不足导致，k8s编排保证服务状态最终一致机制（重新拉起实例）恢复业务，重启大法解决可访问问题。也就是说所有服务都有/ping接口，那服务器上运行的服务ip列表都是需要的。

14.2 再讲依赖列表问题。探测分为端到端的探测（服务互调）和集中式探测（LB类对注册地址的探测）。探测列表，全服务端列表/部分服务端列表。全列表会导致探测量跟调用端成正比，在调用者数量可观的时候，纯探测压力就比较大。毛老师说的子集算法是部分服务端列表，是用C端数量，是保证了服务端随机均衡分布覆盖，同时又不必每一个C端请求所有的S端，降低了服务端探测压力。这个就是个调灵和迟钝的折中，对失败容忍性低的就调短探测时间。健康检查还有种粗粒度的方式就端口探测，网络层的探测，对应用侵入小、消耗小，代价就是没应用级别准确，比如进程夯住就无法发现。

15. 跨服务分页：两个条件在两个服务的情况怎么操作，没有主题库的情况下（插队可耻）

你看能否解决你的问题 1.某个表做下数据冗余可以解决 2.再做一个表做聚合然后搜索

16. 老师，cache更新，你们有个读取处理binlog的服务，那么这个服务怎么保证每次能够读取正确的binlog点位的，是这个服务会每次记录读取binlog点位到consul之类的嘛还是怎么处理的

canal处理binlog服务，然后消息写入kafka，保证消费者处理kafka读取是正确的binlog点位。

17. 处理binlog的这个服务不能多副本部署吧 否则处理binlog会重复吧 如果只是一个服务处理binlog会有性能问题吧 -- cannal同步数据到新的数据库，只是同步下数据，还是会做数据清洗？

一般是单实例部署就可以了，消息接到kafka，由下游的消费者再来处理写入。canal一般是用来同步数据，如果想处理的话，可以自己在消费者那做清洗逻辑。

18. 多集群如何保证各个集群间的数据一致性？

集群按是否存储数据可分为无状态集群和有状态集群

处理请求逻辑，但不存储数据的是无状态集群，无需保障数据一致性

可能处理请求逻辑，但一定存储数据的是有状态集群。分布式CAP理论，AP或CP。CP有著名的一致性算法，Paxos、Raft

毛老师在课程中好像说到这个问题，更新是通过事件广播的，一次不成功就广播多次直到成功为止，也就是最终一致性

19. 如果使用服务端发现模式，healthcheck的性能压力是不是可以减少很多，还需要那个子集算法么？

需要看量级，比如大型的LB集群有无意间healthcheck就会把低配置实例服务打卦，但小集群就没问题，这个是服务端多到一定程度引起的问题。那个子集算法也是客户端多到一定程度引起的问题，所以不要过早关注性能，这个不管是开发和架构里都是适用的，当然设计的时候也要关注边界，把握好这个度不容易。

20. 多集群和单集群多节点比有什么优势

故障域隔离程度不一样。物理的单机故障或逻辑故障，影响的范围是你故障的比例。隔离又分物理隔离和逻辑隔离。抗物理风险高，演变成多数据中心多集群；抗逻辑故障，同数据中心多集群。微服务拆分了服务和存储，一定要注意物理上是否隔离（内核现在隔离cpu、mem很好，但磁盘io和网络io不好）。分割的同时会带来管理上的复杂度，按需取舍。

21. SAAS 平台类的系统，天然支持您说的多租户，我理解的没问题吧？

多租户就是多用户，SAAS产品从设计的时候就是考虑的不同用户如何使用它部分产品或资源。考虑资源隔离、审计、计费。

毛老师在微服务里引进多租户概念是解决资源访问和隔离。

22. 多租户服务是隔离开来了，但是使用的资源是没有隔离的？如果服务存在bug，资源没有隔离会影响生产环境吧？

a. 线上多租户通常指染色、灰度，这个流量通常不会太大，可以在网关配置灰度流量比例

23. 多集群机制有没有做联邦，有没有做多云厂商的多活呢

a. 多云一般建议做多可用区，两个云之间通常没有专线相通，多集群可以在同个AZ里

24. context在父goroutine与子goroutine之间怎么传递？

a. 如果需要控制子goroutine，直接传递进去就可以了

25. 跨服务分页：两个条件在两个服务的情况怎么操作，没有主题库的情况下，比如：订单时间小于 2019 下单人 性别是女| 这样就 是两个服务（一个 订单 一个用户）我理解那个同学想问这种？

26. 多租户的核心是不是就是，给流量打上不同的标签。如果某些服务需要测试的新功能就专注该标签的流量来进行处理就ok了？

27. 请问老师微服务经常用到级联故障，对于这种问题排查有什么可以借鉴的经验么

28. 子集算法没有理解，老师能否在告知下刚才提到的开源代码中哪部分是子集算法。////SRE：Google运维解密 page201

29. 容器是通过macvlan，将容器的网络直接放到物理机所在网络中，通过服务发现拿到IP访问的吗？不通过k8s的service和ingress是吗

30. 对于多集群 的那个cache hit ratio 下降,整个集群全部连接没太听懂 // 缓存击穿

31. 多租户测试，新建的是新的mysql库还是表，还是实例？和生产在同一个server上吗
新建的是 shadow 表，在一个 server 上。

32. 针对API请求的站全链路压测可以使用染色方法，如果是针对资源3消耗类型场景的全链路压测，怎么建设压测环境？（比如B站怎么做视频投稿/视频播放场景的压测）

33. 微服务这块，后面会有相应的实践吗
培训有课表，可以留意一下。

34. 压测的资源隔离怎么做的？机房直接隔离开？

35. healthcheck跟子集算法的关系不太理解能讲更具体一点吗？

可以参考老师课上给出的讲解，简单讲两者都是为了使得消费者让自己尽早的从负载均衡池中摘掉，做到平滑下线。

36. 多租户压测数据是怎么产生？

可以考虑从线上真实数据提取数据做加工（比如去隐私）回灌到生产环境做压力测试。

37. 公司使用java的springcloud 目前是想用golang去慢慢的替换，注册中心使用的eureka请问老师我想把golang的服务注册到eureka应该注意什么。还有后面的逐步迁移有什么需要注意的地方

这个需要毛剑老师来回答一下。

38. 服务发现能否复用业务请求来避免心跳包开销，即业务请求本身也是一种心跳。也就是说客户端调用服务时，客户端本地对每个请求的响应情况做周期性统计，对于网络问题（超时、网络不可达）或者 5XX 等异常，根据异常比例将对应访问节点的权重适当降低。具体实现的时候，可以做到中间件里面，在 RPC 返回前做统计。这种方式是不是更简单，而且还能减少心跳包的开销，也不需要引入子集选择算法。

Heartbeat请求相当于echo，成功的话说明对方一定是活着的，不成功说明对方网络有问题，如果使用业务请求，那么当业务请求失败了，你怎么判断失败是业务本身的问题还是服务器网络问题？另外心跳请求需要定时去发送，业务请求怎么做到？最后业务请求的时间基本都大于心跳请求的时间，这个代替心跳检查给服务器带来的负担太大，浪费计算资源。

39. redis 溃的时候需要，穿透到DB一定是不可接受的吗？如果有些场景允许穿透，应该怎么弄？<如果DB撑不住redis崩溃时的流量，那这种场景就是不可接受的。如果允许穿透那就代码实现了，比如先查询缓存，然后查数据库。>

40. 多集群理解：按照业务划分集群时，比如按照电商，直播，游戏把账号集群划分成3个集群，当电商的账号业务集群宕机；流量会打到直播和游戏，由于直播游戏的缓存中没有电商账号缓存，会直接请求MySQL。为了解决这个问题，后面的进化版本的算法，没太听懂。是不是不管什么业务，调用账号服务时，按照一定的算法，分配到这三个账号服务集群，每个账号集群都保留着所有业务的大部分账号缓存；，这样当一个集群宕机之后，其他两个集群承担流量之后，只会有少部分请求会打到MySQL?这样理解对吗？

41. K8S也有个多租户的概念，这个跟课上讲的多租户是不是有联系？

42. 流量染色、中间件 shadow system 适配等使用中间件的方式来做基础设施处理，会不会导致在推广和升级的时候很困难（毕竟中间件的开发也是有迭代周期的，每次升级或者补充新功能，需要推动所有模块发布），这里为什么不采用 Service Mesh 的无侵入解决方案？如果采用 Service Mesh 需要注意什么？

43. 客户端服务发现不是在同一个name space或pod里起lb那是怎么做直连的，耦合在代码吗

44. 服务发现，消费者，在之前的架构中是怎么分布？消费者是BFF？

45. 刚才说的多集群都是服务的集群，想在了解一下服务后端的 cache 和 db 是怎么架构的？
柠檬：分情况。

1.如果服务的数据是完全隔离的，那就可以使用不同的cache, db. 比如数据根据id hash到不同的集群进行存储，读写。

2.使用隔离的数据，但是底层通过同步机制进行双向同步。

3.使用隔离的数据，但一处为写，其他为读，底层进行单向同步

4.使用一套数据源，如果数据挂了，所有服务也都不可用了，大部分情况下，数据源都是比较稳定的。

46. K8S service 自动发现和外挂的自动发现在应用上有什么区别么（我之前知道的zookeeper是可以有接口维度注册。）

柠檬：

服务注册过程指的是在服务注册表中登记一个服务，以便让其它服务发现。这两者都是这个含义。首先需要服务注册中心，然后需要服务注册，其实是发现和访问。

不同点总结来看：

1.注册粒度不同，一个是进程，一个是pod。

2. 访问方式不同，zk的方式更多是直接ip port访问，而k8s的访问经过层数更多。

3. k8s的模式更为标准化与云原生。zk/etcd/consul/eureka/自研的方式各种各样，k8s比较标准化。

类似于zk的服务注册需要服务自己去往zk的某个目录节点写入进程自己的信息(ip/port)，该目录节点作为服务名或者服务名的一部分。且一般服务A通过zk拿到服务B的某个 ip, port 后直接连接，进行调用。

k8s的service代表的是符合某个selector的一系列pod. k8s的pod是注册在etcd中. 这两个与zk类似. k8s的注册以pod为粒度注册, 且k8s的service访问模式有很多种.

集群内访问模式: 由kube-proxy组件和iptables发现及转发流量.

向集群外暴露 Service: LoadBalancer. 外部的负载监听器监听service的pod变化, 然后配置负载均衡器.

<https://developer.aliyun.com/article/728115>

实际访问链路是什么样的呢? 比如说从集群内部的一个 Client Pod3 去访问 Service, 就类似于刚才所演示的一个效果. Client Pod3 首先通过 CoreDNS 这里去解析出 ServiceIP, CoreDNS 会返回给它 ServiceName 所对应的 service IP 是什么, 这个 Client Pod3 就会拿这个 Service IP 去做请求, 它的请求到宿主机的网络之后, 就会被 kube-proxy 所配置的 iptables 或者 IPVS 去做一层拦截处理, 之后去负载均衡到每一个实际的后端 pod 上面去, 这样就实现了一个负载均衡以及服务发现.

对于外部的流量, 比如说刚才通过公网访问的一个请求. 它是通过外部的一个负载均衡器 Cloud Controller Manager 去监听 service 的变化之后, 去配置的一个负载均衡器, 然后转发到节点上的一个 NodePort 上面去, NodePort 也会经过 kube-proxy 的一个配置的一个 iptables, 把 NodePort 的流量转换成 ClusterIP, 紧接着转换成后端的一个 pod 的 IP 地址, 去做负载均衡以及服务发现. 这就是整个 K8s 服务发现以及 K8s Service 整体的结构.

此处我了解不深入. 请其他助教补充.

47. 微服务之间的调用怎样保证操作的原子性呢? 比如有用户服务、钱包服务、商品服务, 下单之后, 数据需要保证1、钱包扣钱 2、商品下单 3、用户信息更改, 同时完成.

柠檬: 首先大部分服务之间的数据相互独立. 其次大部分情况下, 数据存在不一致也正常. 比如用户头像, 昵称, 很多服务是可以做缓存的. 一旦需要考虑必须要一致的这个问题, 那就是个复杂问题了. 有以下等方法.

最大努力通知型: 基于唯一id号当场重试, 重试多次后失败. 提供接口查询, 校对.

TCC: 用户根据自己的业务场景实现 Try、Confirm 和 Cancel 三个操作; 事务发起方在一阶段执行 Try 方式, 在二阶段提交执行 Confirm 方法, 二阶段回滚执行 Cancel 方法.

本地消息表: 本地事务与本地消息数据表一起插入, 由定时任务发送消息表中的数据到队列中.

事务消息: 本地事务的执行与消息发出去具有一致性, 有专门的事务消息中间件支持, 比如 rocketmq, 其实是上面本地消息表的一种演进.

48. 多租户: 代码层面的服务、数据层面的DB/Cache通过shadow隔离了, 但是资源方面如何隔离, 如何做到全链路压测不影响生产环境的资源呢?

柠檬: 存在全链路压测影响现网的情况, 一般应该在低峰比如凌晨的时候压测. 有的则将压测流量染色转入专门的压测环境, 需要所有的服务均支持.

49. 云厂商的时候的多活, 注意哪些多活, 主要是可用区的多活吧

柠檬: 涉及到数据的话, 多活会很麻烦. 目前还比较少这么强大的组件.

此处我了解不深入. 请其他助教补充.

50. 服务内部调用时, 是否会用到BFF的请求需要一个请求调用多个接口

答: 这应该是一种常态, 即BFF会调用多个服务来完成某一个请求. 本质上来说, BFF是一种特殊的聚合服务.

51. 一个评论服务用户信息部分 是在评论服务聚合还是在 bff聚合?

答: 评论服务不维护任何的用户信息, 它只拥有用户的ID, 那么聚合只能在BFF处聚合, 由BFF去拿到用户的基本信息(如用户名, 昵称), 而后聚合结果之后返回给用户.

52. 兼容的多版本API的实现：是不是每个版本都有个独立的微服务，然后根据API版本打到对应版本的微服务？

答：业界做法是非常多样的。普遍有两种：一种就是你提到的这种，不同的版本直接部署成独立的服务，并且是独立的实例。这种做法的好处是隔离性很强，互不影响，代价是运维复杂；另外一种实践是，不同版本部署在同一个微服务，但是在请求里面会有版本标识，服务器在收到请求之后根据版本来断定应该使用哪个实现。这个需要微服务框架的支持。

如果微服务框架在服务发现过程，就支持版本，也就是，当你调用某个版本的服务的时候，你只能发现该版本的服务，那么从逻辑上来说，不同版本的服务就是彻底不同的服务。而在实际部署中，是否是单独部署，就不太重要了。

53. 续第一个课105题，就是有关CQRS部分，文稿和审核，审核那么复杂为什么不直接把审核部分分离出来？

答：你说的分离是指接口分离还是审核整个分离？一般来说，审核都是一个单独的服务——大公司应该叫做审核平台，负责对外屏蔽掉审核细节。例如是机器审核还是人工审核，机器审核之后要不要人工审核。业务要做的就是审核的内容推过去，而后去审核平台请求一个审核流程就可以。如果涉及到CQRS，我理解的，这里的意思是，和审核平台交互的地方为什么不分离？分离会更好。

54. 关于课程中说的eruka的新老数据按照时间戳进行覆盖，实际上服务器时间是有误差的，那么有一定的概率误删吗？还是说这里问题实践中，发生的概率不那么高？

答：从概率上来说的确是微乎其微。这个问题涉及到分布式系统中一个非常核心的问题，即分布式系统实例时间同步的问题。一般来说，机器都要定期校准自己的时间，所以理论上的确存在这种误差，只不过是微小的。

55. vim和goland推荐哪种？工程化大都推荐Goland，这时候用vim协作会有问题吗？<既然能问这个问题那么就是goland了，vim除非你能很好的控制他，否则还是goland更好>

56. protocol buffer多个微服务间的共享，git submodule 当初考虑过吗？pb定义的代码有版本之分吗？

57. 老师，大数据分析的结果（比如推荐，千人千面）怎么接入当前的微服务体系，

58. grpc 客户端负载，需要调用的服务有多个节点，是每个节点建立一个连接，然后根据负载均衡策略每次选一个用吗？还是选择一个节点只建一个连接，如果是后者，新上线节点如何有流量？

59. 假如我有两个库A和B，A库中要查询id, name, age这三个字段，B库要查询id, use_time这两个字段，同时要满足age > 30, use_time > 100这两个查询条件，同时还要满足分页需求，这样怎么查询？（楼下和我似乎是一样的问题）（加冗余字段可以部分解决，但是如果我的筛选条件更多，需要在B1, B2, B3...多个库中同时做筛选，B1, B2, B3这些库中的ID和A库中的ID是对应的，这种跨服务，多条件，需要分页（获取总页数，当前页数）的查询有什么办法吗）

60. 新旧系统上线，用API网关自动发现服务上线，如果服务上线直接转发请求到新节点，没发现这个服务的直接转发到旧代码，比写一大堆rewrite 好吧？也是一个接口一个接口切如果是这样的话，则需要看是否新旧服务都是用同一个db。如果异构数据库，可能会造成数据混乱，增加对数成本与风险。同构则无所谓。

61. 如何设计多租户的流量标签的自动管理，当有新的标签需要测试时，可以实现自动分发。代码规范上是否需要提前设计好一套规范，每个服务开发时需要遵循这套标识。

需要在 PaaS 平台上进行支持，建基于 k8s 上的标签，建立用户与标签的关联关系。客户端请求时，获取用户的标签，放入请求，从链路中透传。

62. 全链路压测 DB 实例如果和生产环境共用，会不会因为压测负载过高把生产环境搞挂？

建议通过生产 DB 产生一个影子库进行压测。如果使用生产库，会出现脏数据。产品与运营小姐姐可能会发飙。

63. 多集群，互蹭解决热点数据穿透问题，那最终不就变成了不是某个集群针对某个业务的设计初衷么

课上的例子是：原先的账号服务，热点都缓存在服务里头。并且原先账号之间是分集群隔离部署的。比如游戏一个集群，电商一个。当某一天电商集群突然挂了，然后就把流量打到游戏那个集群，会发现玩电商的那波人，在游戏里头的缓存都没有，所以就造成了热点穿透的问题。最后解决的方案是：把所有集群的用户都缓存起来。这样后面即使某个集群挂了，也可以用另外一个集群。

64. 老师好，服务发现的客户端发现模式，是还需要一个注册中心来保存所有服务的真实链接地址，就类似dns解析域名到ip上。那这个注册应该岂不是没有去中心化？谢谢
现在可以选择的开源注册中心，基本上都可以通过部署模型上达到分布式的效果。但不一样的中间件达到的不一样的可用性标准。

65. 平滑退出后，新程序的发布是通过k8s来实现的么？

66. 怎样看待，sre谷歌运维解密中讲到的错误预算

67. A服务起了3个副本，注册到注册中心，B服务调用A服务，向注册中心拿A服务的ip列表，这个拿的过程是B服务自己写代码实现吗？拿到A服务的ip列表后，具体请求哪个ip的负载均衡算法，也是要B服务自己写代码实现吗？

68. consul自带的有健康检查功能，而课上讲的B服务会调用它依赖的A服务的healthcheck接口，这种B服务自己实现检查它所有依赖服务的healthcheck，是为了弥补你们注册中心没有healthcheck这种情况吗？那B服务是不是要定时调用所有它依赖服务的healthcheck接口？

69. service mesh为什么对大多数公司没有收益，是因为：

a. 开销大，是因为进程间通信造成的开销大么

b. 复杂：可能接触的不算多，哪里复杂呢，是要在mesh中做限流、容灾比较复杂么

客户端服务发现+自身实现LB和服务端服务发现，在请求次数上并没有减少吧，只是做到了去中心化吧，因为都是三次请求：

c. 服务端服务发现：请求到LB、LB到服务发现、LB到后端

d. 客户端服务发现+自身实现LB：请求到服务发现、请求到LB、LB到后端

70. Provider服务到Discovery注册后，Consumer服务如何从Discovery获取Provider信息（是Consumer主动拉？还是Discovery推）？如果主动拉需不需要考虑consumer同时发起拉请求可能会使得Discovery处理不过来的问题？如果是推一般采用什么策略（推送的时机）？不同得服务发现中间件在细节上都有细微差异，从性价比上来讲但是都围绕着AP理论。参考lesson2中eureka中实现，consumer-side会在启动时向服务发现中心订阅Provider-side连接信息，为了避免过于频繁得向中心请求压力，选择了30s长连接得patch方式打包这段时间得更新信息，这样平滑了拉取压力，同时有heath check帮助下保证可用（同时也可选不同的poll-period来避免错峰）。

71. 毛老师好，我有两个问题想请教一下

1)、微服务后b站是如何解决分布式事务的？

2)、微服务后要跨多个服务查询数据，是怎么实现之前单个数据库的join、group by、order by功能的？是将各个微服务查询出来的数据放在内存中然后用代码实现join、group by、order by的功能吗？这样做会不会很复杂？

72. API Gateway->BFF->Service：API Gateway也是一个微服务？另外老师说API Gateway里做隔离、限流，负载均衡等可用性，是不是除了API Gateway,下面的像BFF、Service 就只需要关注业务逻辑了，不用管什么限流等了吗？

首先要理解API Gateway (A)、BFF(B)、service (S) 之间的角色定义，比如限流，A是系统服务的入口、B是业务聚合的服务、C是数据源。那么限流可以是为"发言"这个协议做限流，那么B就是"具体哪个UID"发言做限流、C就是为"发言计数信息"做限流。分布式系统中没有统一的答案，也没有准确的规则，一切都需要按照实际场景因地制宜。

73. 请问微服务之前怎么实现事务处理的？

74. 对于使用多个节点来缓解压力，总需要一个负载均衡来分配，那这个负载均衡就需要承受全部压力，即使这个负载均衡也多节点，那也需要一个类似LVS的东西，来分配给多个负载均衡，它就要承受所有的压力，它如何承受得住所有压力的呢？
82. grpc的健康检查是服务提供者与调用者的检查，服务调用者也会向eureka发送健康检查，eureka也会定期发送检测到provider者检测服务可用性，我这样理解对么？
83. 大量的Consumer与Provide进行healthcheck，如果使用子集算法对Provide节点取子集不应该会更危险吗？
85. 分层结构中的API-gateway可以理解为就是k8s中的

毛剑老师来答：

75. 毛老师，能不能推荐几个微服务实践项目代码，可以用来学习
76. 毛老师，源码阅读有没有好的切入点建议，还是直接硬啃
77. 老师可以讲讲grpc最新用的xds嘛？网上都没找到例子，只有协议描述，但是没有get到点。
78. B 站只用了 k8s 的容器编排吗？还用了其它哪些功能？
79. 老师可以讲一下为啥不用istio？istio 也提供了基于envoy实现的sidecar,可以提供染色，熔断，金丝雀发布，而且是非侵入式的，改改配置文件就能实现，它不香吗？
80. 毛老师，可以分享你们标准化启动goroutine是怎么做的吗？谢谢
81. b站数据库db方面数据量最大的表有存多少数据，查起来会不会比较慢如果没有分表的话
82. 实战项目，比如在这一节最后讲的多租户测试环境，有没有一个实际的代码环境去演练下
83. 公司使用java的springcloud 目前是想用golang去慢慢的替换，注册中心使用的eureka请问老师我想把golang的服务注册到eureka应该注意什么。还有后面的逐步迁移有什么需要注意的地方
84. 为什么要多集群要独占缓存然后订阅binlog，多集群共享一个缓存不行么。独享的话如果多集群质检上了redis怎么办