1.比如说在一个游戏行业里面，或者说一个典型的游戏架构，在云资源层面，

就是云的架构层面。应该注意哪些点，或者怎么样才能设计一个符合这个游戏

需求的一个架构。比如说它可能是有一个小的游戏，或者说是一个很大的像原神

这样的，或者它可能是一个很小的页游或者端游或者什么样。那么在不同场景下，

它的架构会有什么区别？然后这块的考虑因素是什么？

答：在设计一个满足游戏需求的架构时，需要考虑以下几个方面：

游戏类型和规模：小型网页游戏或手机网络游戏，一般不需要太复杂的架构，可以直接使用云服务器ECS，数据库RDS等基本服务来搭建。而对于大型网游如原神，则需要更为复杂的分布式架构，包括多台服务器协同工作，可能需要使用到弹性伸缩、负载均衡等高级功能。

用户访问量：如果预计有大量的用户同时在线，那么需要考虑使用负载均衡SLB来进行流量分发，以避免单一服务器过载。同时，还需要考虑到高并发情况下的数据一致性问题。

数据处理能力：对于大数据量的游戏，例如拥有大量用户数据和行为数据的MMORPG，需要使用高效的分布式数据库和存储服务，如DRDS和OSS。

网络安全：任何在线服务都需要考虑网络安全问题，特别是游戏这样的高价值目标，更是黑客的重点攻击对象。因此，需要使用WAF等防火墙设备，以及DDoS防护等手段来保护服务器安全。

总的来说，对于不同的游戏场景，其架构设计的关键点也会有所不同，但是都要围绕着如何更好地服务于玩家这一核心目标进行。

2.比如说对假如说我们现在这款游戏刚上线，它可能这个在线用户数或者说并发数。可能刚上线他会没那么多，它可能初始是一个1000QPS的，那可能随着我的运营做的比较好，这个游戏也做的比较好，它可能会变成1万甚至更多的QPS对于一个这个架构的演进。或者说从一个1000QPS的系统到了一个1万QPS的系统，它应该是怎样的一个过程，或者说怎么设计，什么样的架构，是更符合这种场景的，是一套架构应对这两种场景吗？还是不同的场景下面要调用不同的架构？

答：对于一款游戏来说，从1000QPS增长到10000QPS是一个比较大的挑战，因为这不仅涉及到硬件资源的扩展，

还包括软件架构的设计和优化。

初始阶段：当游戏刚开始上线，预计用户量不会太大，可以选择使用相对简单的架构，例如使用一台或几台云服务器ECS，

搭配MySQL数据库，以满足初期的需求。

中期阶段：随着用户量的增长，可能需要升级硬件资源，例如增加云服务器的数量，并使用负载均衡SLB进行流量分发。此外，也可以开始考虑进行一些软件架构的优化，例如使用缓存服务Redis来缓解数据库的压力。

后期阶段：当用户量进一步增长，可能需要采用更为复杂的技术方案，例如分布式数据库和存储等技术，以保证系统的稳定性和可扩展性。

在这个过程中，可以使用一些工具和方法来监控系统的运行状况，以便及时发现并解决问题。例如，可以通过监控系统的

CPU、内存、磁盘I/O等指标，来判断是否需要进行硬件升级；可以通过分析系统的访问日志，来了解用户的使用习惯和热

点问题，以便于进行相应的优化。总体来说，一个好的游戏架构应该能够在应对不同规模的用户量的同时，保持良好的稳定性、可扩展性和可用性。

同时，也需要根据业务的发展变化，不断地进行调整和优化。

3.我们用云的环境弹性伸缩来解决一些风险流量的一些问题。但是用弹性探索也需要一些采用分布式的系统，需要稳定性的问题。

那我想问一下，比如说在使用负载均衡这种场景下，在高可用这个逻辑上面，我们要怎么做才算是一个比较好的情况？

答：负载均衡策略：合理的负载均衡策略也是提高系统可用性的重要手段之一。例如，我们可以根据服务器的实际性能和负载情况动态调整分配给它们的任务数量，避免某一部分服务器过载而导致整体服务质量下降。

系统日志监控 : 检测系统的错误日志并快速响应。

容灾和备份设计：通过容灾和备份设计来增强系统的可靠性。例如，可以设置多个数据中心，当其中一个中心出现问题时，可以快速切换到另一个中心继续提供服务。

4.你知道我们的这个负载均衡，它是一个地域级别的产品，它做不到跨地域。那你是怎么做到负载均衡的入口，同时在不同的定位都可用。比如说北京在杭州，在上海都有，你的负责，都有你的这个机器，你是怎么办？到这个流量的均衡，然后同时大家又都在一个全局的游戏中。

答：使用SDN（Software Defined Networking）或者Overlay网络技术，将分布在不同地域的服务器组成一个虚拟的网络。在这种情况下，你可以使用传统的负载均衡器来进行流量分发，而无需担心地域之间的通信问题。这种方法的优点是可以提供更好的性能和灵活性，但实现起来比较复杂，需要专业的网络知识和技术支持。

5.那也就是说你不同定义里面的机器是不互通的，就是只能跟着一个组的。那假如我今天我注册的时候，我是在上海注册的。然后我这个用户出差跑到了北京。这个时候他如果想登录这个游戏，他应该去哪？他只能去上海吗？

答：还是应该在上海，我们设置的是若玩家连续30天都在上海登录，才会改变他的服务区，这样保持一定阈值时间，避免增加系统操作压力。

6.这个其实是我刚才问了这一系列问题，想要想想就是想表示的一件事情，就是你的稳定性，和你的性能，还有你的成本，这之间是需要去综合考虑。的。这个是我们要在做架构设计的时候去考虑的。你们的游戏部署在阿里云的哪些region上？好，你们在部署杭州北京两个region上，假设现在有一个情况，我在西藏，假设我在西藏，我也想玩这个游戏。然后我发现这个网络总是不太行，就动不动就垮，然后一抖我就下线或者掉线什么的就不太好。那这种场景下怎么优化一下呢？

答：可以使用以下方法：

使用CDN(Content Delivery Network)服务：CDN可以在靠近用户的位置提供内容分发服务，减少数据传输的距离和时间，进而提升访问速度。例如，阿里云的CDN节点遍布全国各地，包括西藏地区，可以帮助您改善用户体验。

使用边缘计算：您可以考虑使用边缘计算连接西藏与您服务器所在的地区，减少网络传输距离，降低网络延迟。

异步更新和数据压缩：对于游戏客户端而言，可以采用异步更新、数据压缩等技术，减少客户端与服务器之间的数据传输量，减小网络负担，提高响应速度。

7.假设说现在这个西藏的时间节点覆盖不太好，因为你知道西藏的这个基础设施不太行的运营商也不太行。但但是又有那么一些，你可以假设它不是一个C端用户，或者说可能是你的一个VIP用户，或者他的一个氪金的玩家。Anyway他就很重要，这个时候你要应该怎么去做。

若这名玩家价值很高，可以使用以下方法来提升他的游戏网络体验：

1-分布式部署，直接在阿里云在西藏的适当节点为此玩家设置专门的VPC

2-VPN专线连接，确保网络的稳定性

8.假设现在韩国的玩家必须要接入。你现在在您现在在国内的北京或者上海的系统来做。在这种情况下，怎么保证韩国玩家的访问速度？这个时候怎样处理好跨境的问题？

答：使用全球加速， 阿里云提供了全球化访问的产品, 如Global Accelerator (GA), GA在全球范围内提供了高速稳定的网络链接, 可以为您的用户提供低延迟的游戏体验

9.CEN和全球加速有什么区别？

答：GA全球加速不需要在客户那一侧部署什么云资源的，就是只需要给人家一个上车点，他就直接可以访问了。但是CEN是连接了2个VPC PC的，连接2个VPC的，你一定要在韩国有个VPC, 就是你要在韩国，在阿里云region上面部署一些资源，你才需要去用云企业网去打通韩国和北京。那就变成说今天你如果用云企业法的方案的话，就是你不再只是北京和杭州两个region，你要在韩国也要有一个region部署ECS部署数据库，所以CEN要相对GA全球加速更加稳定些。

10.多服游戏架构和单服游戏架构有哪些因素需要注意和关注的，来提升整体的成为更好的游戏系统？

答：在设计多服游戏架构时，需要关注的因素有：

数据同步: 在多服架构中，游戏世界和数据同步是非常重要的，因为不同的服务器可能有不同的游戏进度。游戏世界同步机制可以保持

所有的服务器上的世界状态一致，让用户能够自由的在各个服务器间切换。

扩展性: 多服架构比单服更容易扩展和调整容量，因为它可以把压力分散到多个服务器上，而不是全部集中在一个服务器。

管理复杂度: 多服增加了管理复杂度，需要维护更多的服务器，保持所有服务器的一致性和协调各个服务器之间的操作。

对于单服游戏架构，关注的因素包括:

性能和稳定性：需要确保单个服务器能够承受大量的用户同时在线，同时也要确保服务器的稳定性和安全性。

扩展性: 单服架构相比多服来说很难快速扩大容量，当用户人数过多时可能会导致服务器崩溃。因此，在设计之初就要考虑到扩展性，

如使用可扩展的数据库和存储系统。

总的来说，无论是多服还是单服的游戏架构，关键是要满足用户的需求，提供稳定、快速、无缝的游戏体验，同时也需要适应游戏的

发展和变化，做出适当的优化和改进。

11.假如说现在你是北京、上海和韩国首尔三个region,那么你可以是一个服务区，有三个region；你也可以是两个服务区。比如北京、上海是一个服务区，然后韩国是一个服务区。这两种情况下有什么区别？对你架构设计有什么影响？

答：北京、上海和韩国首尔三个region属于一个服务区，对游戏数据库压力比较大，增加了运营和维护，增加管理难度。相比之下，

第二种方案减少了运营和维护的工作量，分开的服务区可以让你更灵活地进行扩展。

12.不同服的情况下，平台级的更新怎样做更好？效率会更高呢？

答：可以使用阿里云运维编排服务（Operation Orchestration Service，简称OOS）通过创建模板-创建执行-执行结果这一套流程，跨地域进行事件驱动的批量操作、定时运维任务来实现平台级的更新

14. 比如说已经确定好整个游戏架构，我们设计出来了，然后甚至都已经在一些POC的阶段，在这个做测试了。我们怎么做一些性能测试或者一些这种performance的一些evaluation。来看一下，我要不要做一些调整，这个测试应该怎么做。

答：在游戏上线之前，通常需要执行一系列的性能测试以评估架构的有效性和可行性。以下是一种可能的测试流程：

设定测试目标: 首先明确测试的目标是什么，例如游戏是否能在峰值时段满足预期的 QPS (每秒查询次数)，是否能满足预计的并发用户数量等等。

设计测试案例: 根据设定的目标设计测试案例，包括但不限于压力测试、功能测试、安全性测试等。

进行测试: 在模拟环境下进行测试，并收集相关的指标数据。

结果分析: 分析测试结果，看是否达到预期的目标，并找出可能导致问题的地方。

调整与优化: 根据测试结果对游戏系统进行调整和优化。

15.PTS是一个挺好用的工具，然后利用PTS的话应该怎么做才能比较好好的看出，或者说评估一下这个现在是我们觉得是ok的还是怎么样。

答：对于利用PTS做性能评估，可以有一下一些注意事项

首先明确测试目标: 明确你要做什么类型的性能测试 (例如压力测试、并发测试) 和期望的结果(例如多少并发量)

观察结果: 执行PTS任务后，观看系统的反应并获取相关性能指标。

结果分析: 根据PTS的结果进行分析，并确定是否有潜在的问题。

1. 如果说我们现在发现在这个过程中，我的这个transaction就是数据库的transaction有点慢。老卡，对对对，我现在这个半天应该是可能很多操作是原子性的事物，然后就可能卡住了。如果这种情况下，我们要怎么优化一下，或者说从哪些方向上去考虑？

答：当发现数据库事物慢的时候，可以从以下几个方面入手进行优化：

读写分离：读取密集型的应用程序可以使用主从复制和缓存，将读取操作放到从库或缓存上，减轻主库的压力

SQL查询优化：查看慢查询日志，找出导致性能瓶颈的问题查询语句，并针对性地对其进行优化

表结构优化: 调整字段的数据类型，增加索引等

MySQL参数调优: 调整MySQL的相关参数，例如 innodb\_buffer\_pool\_size等

数据库分区 : 把大表按照某种规则分割成若干个小表，可以改善读写速度

数据库扩展：垂直扩展或水平扩展数据库集群

使用数据库中间件 : 放在主从复制的基础上，可以把主库变成读写分离模式, 或者实现Sharding

磁盘优化: 检查磁盘IO，查看是否符合需求

17.怎样判断是网络带宽问题？

以下方法可以判断是否是网络带宽的问题：

ping测试: 如果在高峰期访问网站时，ping值较高，可能是网络带宽问题。使用ping工具，发送数据包到目标主机，观察网络延迟和丢包情况。

带宽测试: 用专门的网络带宽测试工具进行网络带宽测试，查看实际带宽利用率。

网络监视工具: 使用网络监视工具检查带宽使用情况。

18.在监控方面，怎样做一个比较完整的可视化？

主要有一下数据需要直接可视化监控

系统资源 : CPU、内存、硬盘空间、网络流量等硬件资源使用情况

游戏进程状态 : 进程运行状态、是否有挂起或阻塞等异常情况

数据库性能 : SQL执行性能、事务性能等

系统事件 : 用户登录成功率、注册成功率等重要系统事件

19.确定所要可视化显示的监控指标的逻辑是什么？

要看核心的业务链路，那这个场景它可能进来的时候走的是CDN，然后走了负载均衡，然后走到ECS, 然后后面可能接了某个mysql或者是某个其实是应该按业务链路的形式来做可视化的设计。就这样一条最重要的链路里面，在这个链路里，各个环节的资源现在是一个什么情况？

因为里面我看到其中有一台ECS CPU, 比较有可能会马上就去看一下我的分发业务有没有受到影响。比较有可能会马上就去看一下我的分发业务有没有受到影响。其实可视化的核心可视化的核心不是给人看的，可视化的核心在我看来是要尽量降低问题发生的概率，以及问题发生之后能够缩短这个问题恢复的时长。