Q1:分布式系统的定义和目标?

【参考第一章】

分布式系统是指一系列独立计算机的集合，对外呈现成唯一且统一的系统。分布式系统的目标有四个，分别是透明性、开放性、资源可达性以及可伸缩性，其中透明性指的是隐藏底层细节使得用户使用的时候就像是在一台独立计算机一样，开放性指的是忽略底层异构，整个分布式系统可以和外界其它开放系统通信互联。

Q2：分布式系统集中式、非集中式和混合式三种组织形式的含义?

【参考第二章】

- 集中式：存在某个中央节点提供特定服务，比如Basic C/S Model
- 非集中式：不存在某个类中央节点，每个节点的功能都是类似的或者对称的，比如P2P
- 混合：P2P可以和一些集中式的特点相互结合

Q3：分布式系统策略与机制分离的理解；并举出一个应用实例?

【感觉上是参考第二章】

经评论区指正，更新了题目

理解：

> 在分布式系统中，将策略与机制分离是一种设计原则，旨在提高系统的灵活性、可维护性和可扩展性。它是将系统中的决策逻辑（策略）与执行逻辑（机制）分开，使得两者可以独立演化和调整。
> 具体来说，策略是指系统中的业务逻辑、决策规则或配置参数，用于指导系统的行为。而机制是指实现策略的具体算法、协议、数据结构或技术实现，用于执行策略并实现系统功能。
> 通过将策略与机制分离，可以带来以下好处：
> 灵活性：策略和机制的分离使得可以独立地修改和调整系统的行为，而无需修改底层的执行逻辑。这样可以更容易地应对业务需求的变化，而无需重构整个系统。
> 可维护性：策略和机制的分离使得系统的维护更加容易。当策略需要变更时，只需修改相应的策略部分，而无需触及底层的机制。这简化了系统的维护和调试过程。
> 可扩展性：通过将策略和机制分离，可以更容易地引入新的策略或机制，从而扩展系统的功能。可以根据具体的需求，选择合适的策略和机制进行组合，以满足不同的应用场景。
> 重用性：策略和机制的分离使得可以在不同的系统或模块中重用相同的机制，以实现相似的功能。这样可以避免重复开发和维护相同的执行逻辑，提高代码的复用性和可维护性。
> 总之，将策略与机制分离是一种良好的设计原则，它通过解耦系统的决策逻辑和执行逻辑，提高了系统的灵活性、可维护性和可扩展性，使得系统更易于开发、维护和演化。

实例：

> 一个典型的应用实例是分布式缓存系统，比如使用策略和机制分离的方式实现一个分布式缓存系统。
> 在这个实例中，策略可以包括缓存的淘汰策略（如LRU、LFU等）、缓存的失效策略（如基于时间或基于事件的失效机制）、缓存的数据复制策略（如一致性哈希等）。这些策略决定了缓存系统的行为和特性。
> 而机制则包括缓存的数据存储结构（如哈希表、分布式数据存储等）、数据传输协议（如基于TCP的通信协议）、数据一致性的实现（如使用分布式一致性算法）、缓存节点的自动发现与选举机制等。这些机制负责执行缓存系统的策略，实现具体的功能。

Q4：代码迁移与虚拟机迁移的优劣?

【参考第三章】

代码迁移的优点：灵活性和可控性，一方面代码文本比较轻量，另一方面可以使用现代版本控制功能比如git，跟踪代码变换历史，使得迁移更加容易和安全。

代码迁移的缺点：兼容性和性能问题。一方面没有考虑硬件环境，有可能导致代码和新平台不兼容的问题，增加开发难度，另一方面不同的硬件可能导致代码性能下降

虚拟机迁移的优点：易于管理和维护，迁移时不需要考虑硬件和环境细节。

虚拟机迁移的缺点：性能开销较大，比较复杂。由于虚拟机需要模拟硬件和操作系统，因此虚拟机迁移可能会引入一些性能开销。此外，虚拟机迁移还需要处理一些复杂的问题，如网络配置、存储管理和虚拟机的生命周期管理。

Q5：客户端没有收到来自服务器的响应，可能的情况有哪些？并给出可能的解决方案?

【第四章】

客户端请求消息丢失

- 客户端设置一个timer超时重发

服务器在收到请求信息后崩溃

- 等待服务器重启后重发，保证RPC至少被执行一次
- 立即放弃并汇报错误，保证RPC至多被执行一次
- 什么也不做，语义有客户端自己保证

服务器传回的消息丢失

- 客户端置一个timer超时重发，对于不幂等的请求，为请求分配序列号，服务器区别不同的请求

客户端不能找到服务器

- 使用返回值-1/exception/signal等代表异常

Q6：在观看直播的时候存在卡顿，请使用 4 种方法来进行优化

【第四章】

（不确定）

PPT上给了三个提高流数据通信的方法：

利用分区服务为不同类型的数据提供服务

利用缓冲区减少延时抖动

交错传输降低丢包影响

gpt给出的答案：

1. CDN（内容分发网络）：使用CDN可以将直播内容分发到离用户更近的边缘节点，减少数据传输的延迟和拥塞。CDN采用分布式架构，将内容缓存到离用户最近的边缘服务器上，用户可以从就近的服务器获取直播内容，减少了数据的传输距离和网络拥塞的可能性，从而提高观看体验。
2. 负载均衡：通过负载均衡技术，将用户的请求分发到多个直播服务器上，实现资源的平衡利用。负载均衡可以根据服务器的负载情况、网络状况等因素，智能地选择合适的服务器来处理用户的请求，从而提高系统的整体吞吐量和响应速度，减少卡顿情况的发生。

3. 前端缓存：在直播系统的前端引入缓存机制，可以缓存一定时间内的直播内容，减少后续用户的请求对后端系统的压力。通过缓存，用户可以从本地缓存中获取直播内容，减少对后端资源的依赖，降低延迟和提高观看体验。
4. 弹性扩展：当直播系统的负载增加时，可以通过弹性扩展来提供更多的资源。分布式系统中的弹性扩展可以通过增加更多的直播服务器、调整资源分配策略等方式来应对高负载情况。这样可以满足更多用户的需求，减少系统的瓶颈和卡顿现象

Q7：为啥要使用动态绑定？动态绑定的缺点是什么？

【第四章】

动态绑定使得客户端可以定位服务器，灵活；可以支持多个服务器提供同一个接口，从而实现负载均衡、容错以及协助认证；binder可以确保客户端和服务器使用同样版本的接口

缺点：导入导出接口产生额外开销；binder在大型分布式系统中可能成为瓶颈

Q8：每个节点有自己的本地时钟，设计一种协议，使得节点的时间等于节点时间的平均值？

【第六章】

Berkeley算法：

master服务器发送服务器到其它所有的slave服务器，询问其时钟值；

其它机器做出应答；

master服务器接收到所有slave服务器时间后，计算时间均值，然后将值回填至所有的服务器，其中包括master服务器

Q9：假设有四个进程，设计一种尽可能快的分布式互斥访问协议，并解释其中某一节点故障时系统为何还能正常工作？

【第六章】

集中式相互排斥协议：

选取一个进程作为协作者，由协作者调度对共享资源的访问；

当一个进程要访问共享资源，需要向协作者发送请求信息，说明其要访问的资源；

如果当前没有其它进程访问资源，协作者就发送准许应答消息。

当其中一个节点故障时候，如果不是协作者进程，则不影响其它进程和协作者交互来实现互斥访问；如果是协作者出现问题，可以重新在没有故障的进程中选取新协作者

gpt的回答：

> 冗余节点：在集中式相互排斥协议中，可以设置多个冗余节点作为备份。这些冗余节点可以立即接管中心节点的角色，接收和处理其他节点的请求。如果其中一个节点发生故障，系统可以自动切换到备份节点，确保系统的连续性和正常工作。容错设计：在设计集中式相互排斥系统时，可以采用冗余和容错技术来保证中心节点的高可用性。例如，使用冗余的硬件设备、备份服务器、故障转移技术等，以确保即使中心节点发生故障，系统仍然能够快速切换到备份节点。恢复机制：如果中心节点发生故障，系统可以通过故障检测和恢复机制来检测故障并采取相应的措施。例如，其他节点可以通过心跳检测或超时机制来检测中心节点的故障，并触发故障恢复过程，例如选举新的中心节点或启动备份节点。临时禁止访问：当中心节点发生故障时，其他节点可以采取一些措施来临时禁止进程访问临界区，以防止系统出现不一致或冲突的情况。例如，节点可以等待中心节点恢复或切换到备份节点后再重新请求许可。

Q10：分布式复制（多副本）有什么优劣?

【第七章】

优势:

- 可靠性：可以避免单点故障
- 性能：在数量和地理区域上的可扩展性

劣势:

- 复制透明度
- 一致性问题：更新成本高昂；可用性可能会受到影响

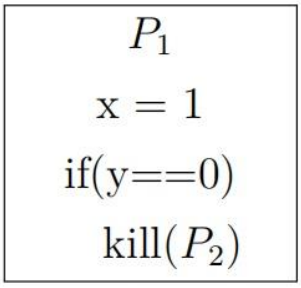
Q11：为什么说 Two-phase-commit 是阻塞性的?

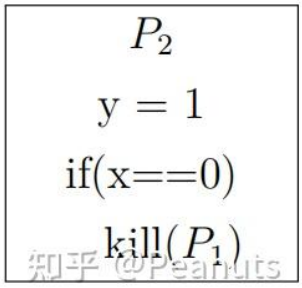
【第八章】

1. 准备阶段（Phase 1）：协调者向所有参与者发送准备请求，并等待它们的响应。参与者会执行事务的准备操作，并将准备就绪的状态（prepared）或者无法准备的状态（aborted）发送回协调者。
2. 决策阶段（Phase 2）：协调者根据所有参与者的响应情况做出决策。如果所有参与者都准备就绪，协调者发送提交请求给所有参与者，并等待它们的确认。如果有任何一个参与者无法准备就绪或者出现错误，协调者发送中止请求给所有参与者。

尽管 Two-phase commit 协议提供了原子性的保证，但它是阻塞性的，即在执行过程中可能会发生阻塞的情况。这是因为在决策阶段，协调者需要等待所有参与者的响应，这可能包括网络通信延迟、参与者故障或者其他因素的影响，导致等待时间不确定。如果其中一个参与者无法响应或者发生故障，协调者会一直等待，直到超时或者其他容错机制触发。

Q12：现有进程 *P*1, *P*2，如果它们遵循顺序一致性，可能产生什么结果？并说明原因。如果它们遵循 FIFO 一致性，可能产生什么结果？并说明原因





【第七章：这题存疑】

顺序一致性

x=1;y=1;if(x==0)kill(P1);if(y==0)kill(P2);两个进程都成功执行

y=1;x=1;if(y==0)kill(P2);if(x==0)kill(P1);两个进程都成功执行

FIFO一致性

x=1;y=1;if(x==0)kill(P1);if(y==0)kill(P2);两个进程都成功执行

x=1;if(x==0)kill(P1);y=1;if(y==0)kill(P2);两个进程都成功执行

y=1;x=1;if(y==0)kill(P2);if(x==0)kill(P1);两个进程都成功执行

y=1;if(y==0)kill(P2);x=1;if(x==0)kill(P1);两个进程都成功执行

Q13：基于客户的一致性协议与基于数据的一致性协议的区别是什么？一个 Web 系统，如何设计多副本模型，选择哪种一致性协议比较好？原因是什么?

【第七章】

区别:

基于数据的一致性协议，数据的共享是通过分布式共享存储器、分布式共享数据库或者分布式共享文件系统实现的；

基于客户的一致性协议相比较而言具有更加宽松的一致性，只考虑最终副本的一致性；在没有任何进一步的更新下，所有的副本都收敛到彼此相同的副本，只需要保证更新将被传播；如果用户总是访问相同的副本则是简单的，如果用户访问不同的副本则会出现问题。

基于客户的一致性模型为单一的客户提供一致性保证，并不能为不同用户的并发性访问提供一致性保证。

选择以及原因:

【chatgpt+phind】

选择基于客户的一致性模型更好，因为相对于基于数据的一致性协议，基于客户的一致性协议通常可以提供更好的性能，因为它允许更高的并发性和较低的延迟。Web系统一般是一个对实时性要求较高，对数据一致性要求相对较弱的场景。所以选基于客户的一致性协议比较好。

> phind的回答：在选择使用基于客户的一致性协议还是基于数据的一致性协议时，需要考虑你的Web系统的特性和需求。基于客户的一致性协议，例如Paxos和Raft，通常用于处理分布式系统中的写操作。这种协议的优点是可以在分布式系统中保证所有节点的一致性，即使在网络分区或者节点故障的情况下也能保证所有节点看到的数据是一致的。然而，这种协议的缺点是在处理大量写操作时，可能会导致系统性能下降，因为每次写操作都需要在所有节点之间进行通信。基于数据的一致性协议，例如Cassandra和DynamoDB，通常用于处理分布式系统中的读操作。这种协议的优点是可以在保证局部一致性的同时，提高系统的读写性能。这是因为这种协议允许数据在不同的节点之间存在一定的延迟，只要保证在给定的时间内，用户可以读到最近的数据即可。然而，这种协议的缺点是在处理写操作时，可能会导致数据的不一致性，因为在写操作完成之前，用户可能会读到旧的数据。总的来说，如果你的Web系统需要处理大量的写操作，并且需要在所有节点之间保持数据的一致性，那么使用基于客户的一致性协议可能更合适。如果你的Web系统需要处理大量的读操作，并且可以接受一定的数据不一致性，那么使用基于数据的一致性协议可能更合适。

Q14：一个 K 容错系统，每个节点返回结果是正确的，如果节点返回的结果是错误的，需要多少个节点保证容错？解释原因，并设计一种协议，解决这种容错问题?

【第八章】

原因：需要至少2k+1个进程才能保证，因为最坏的情况下，K个失败的进程意外产生相应的应答，剩余k+1个进程产生相同的正确应答，这样仍然可以相信多数进程的回答

算法（拜占庭算法）：

1. 每个节点告诉其它n-1个节点自己的value
2. 将每个节点搜集的信息以向量形式保存
3. 每个节点将自己的向量发送给其它所有节点
4. 每个节点检查新收到的向量的第i个元素，把多数值保留下来得到一致性结果

Q15：虚拟化技术解决的问题是什么？虚拟化技术与云计算的关系是什么？

【讲座】但是没有找到当年的PPT，这边结合搜索到的结果

虚拟化解决的问题：

- 资源利用率：虚拟化技术可以将一台物理机划分为多个虚拟机，每个虚拟机可以独立运行不同的操作系统和应用程序。通过虚拟化，可以更好地利用物理服务器的资源，提高资源的利用率。
- 灵活性和可拓展性：虚拟化技术可以将多个物理机组成一个资源池，可以灵活地分配和管理虚拟机，根据需求对虚拟机进行扩展或收缩。这样可以使分布式系统更加灵活，并且能够根据负载的变化进行动态调整，提高系统的可扩展性。
- 高可用性和容错性：通过虚拟化技术，可以实现虚拟机的迁移和复制，当一台物理机发生故障或需要维护时，可以将其上的虚拟机迁移到其他正常运行的物理机上，保证系统的高可用性和容错性。
- 简化管理和部署：虚拟化技术可以提供统一的管理界面和工具，对虚拟机进行集中管理和监控。同时，可以通过虚拟机模板和快速部署功能，简化系统的部署和配置流程，提高管理效率。

虚拟化技术和云计算的关系：

虚拟化技术是云计算的基础，通过虚拟化技术可以提供云计算所需的灵活性、弹性和效率。云计算则是基于虚拟化技术构建起来的一种计算模式，为用户提供各种计算和服务。

具体来说，虚拟化技术通过将物理资源（如服务器、存储和网络）划分为若干虚拟资源，使得多个虚拟机可以在同一台物理机上独立运行不同的操作系统和应用程序。这样，云计算平台可以在这些虚拟机上提供多种服务，如计算、存储和网络服务，给用户提供弹性、灵活的计算环境。

云计算提供的服务包括基础设施即服务（IaaS）、平台即服务（PaaS）和软件即服务（SaaS）。而虚拟化技术是实现这些服务的基础，通过虚拟化技术可以实现资源的共享、按需分配和弹性伸缩，从而提供高效的云计算服务。

此外，虚拟机（VM）是云计算中的一个重要概念。在云计算环境下，用户可以通过虚拟机实例来获得所需的计算资源，并根据需求按照付费模式进行使用，从而实现了资源的弹性分配和管理。