

# 分布式系统考点

标红的为2024年涉及到的考点

## 2022年

### 1.分布式系统的定义，为什么要使用分布式？

答：分布式系统是若干独立计算机的集合，这些计算机对于用户来说就像是单个相关系统。

为什么要使用分布式：

- 经济 (economics)：微处理器比大型计算机更便宜
- 速度 (speed)：分布式系统比单个大型计算机拥有更强算力
- 固有分布式 (inherent distribution)：一些应用在涉及到空间上分散的机器
- 可信赖性 (reliability)：如果一台机器故障，整体系统仍然能正常运转
- 增量增长：可扩展性

### 2.分布式系统的类型

答：1.分布式计算系统：关注高性能计算和资源共享，又分为集群计算系统和网格计算系统。比如超级计算机和HPC。

2.分布式信息系统：侧重于信息存储、管理和检索。比如数据库系统或文件系统。

3.分布式普适系统：以人机交互为核心。比如物联网或云计算。

### 3.TCP和UDP分别位于网络中哪一层？使用TCP和UDP进行RPC通信分别有什么好处？

答：位于传输层 (transport layer) 。

TCP的好处：

- 可靠传输
- 自动处理流量控制和拥塞控制

UDP的好处：

- UDP是无连接的协议，低延迟
- 没有流量和拥塞控制，吞吐量更高

## 4.分布式系统的三层client-server架构是什么？

答： 1.User-interface Layer：包含应用程序用户接口单元  
2.Processing Layer：没有特定数据，但是包含应用程序功能  
3.Data Layer：包含客户端希望通过应用程序组件操作的数据

## 5.代码迁移的强迁移和弱迁移的含义分别是什么？

答： 强迁移指将整个对象从一台机器移动到另一台机器，包括各种组件和执行状态等；  
弱迁移仅迁移代码和数据段。

## 6.Raft三个部分的过程。

答： Raft算法可以分为三个主要部分：领导者选举、日志复制和安全性。下面是这三个部分的简要说明：

### 1. 领导者选举：

- 在Raft算法中，集群中的节点会随机发起选举，如果一个节点发现自己没有收到领导者的心跳信号，它会转变为候选者状态，发起选举请求。
- 候选者会向其他节点发送请求，要求它们投票给自己。如果候选者获得了超过半数节点的投票，它就成为新的领导者。
- 一旦选举成功，新的领导者会开始发送心跳（AppendEntries RPC）来维持领导地位，防止其他节点再次发起选举。
- 如果当前领导者失败或集群出现分区，Raft会再次进行选举，确保集群始终有一个有效的领导者。

### 2.日志复制：

- 领导者负责将客户端的请求（通常是状态变化的请求）写入日志并复制到集群中的所有节点（包括领导者自己）。
- 领导者首先将日志条目添加到自己的日志中，然后将该条目通过AppendEntries RPC发送到追随者节点。
- 追随者节点将日志条目添加到本地日志，并向领导者发送确认。
- 领导者会等待大多数节点的确认，才能将该日志条目提交（commit）。一旦日志条目被提交，领导者会通知所有节点，更新其状态机。

### 3.安全性：

- 领导者的唯一性：任一时刻，集群中只有一个领导者。
- 日志一致性：如果一个日志条目在某个节点上已经被提交，则该条目会被复制到所有后续的节点中，且它的索引和任期号是唯一的。
- 事务的顺序性：所有日志条目都按照严格的顺序进行提交，避免了由于日志条目不同步而导致的数据不一致问题。

## 7.集群训练大模型面临的挑战有哪些？（讲座题）

2024年考点具体不太一样，考的是大模型并行训练有哪些方法、以及大模型训练中通信和普通的分布式集群通讯相比有哪些新特点

- 分布式数据并行性：大模型通常需要处理巨量的数据，分布式训练需要有效地分割数据并分配给不同的计算节点。
- 模型并行性：当模型过大，无法在单个节点上存储时，模型的不同部分必须分布在不同的节点上。
- 通信瓶颈：在分布式训练中，节点之间的通信（例如，梯度同步）是一个关键问题。
- 故障容忍性：在大规模集群中，节点可能会发生故障或失效。如何保证在节点失效时，训练能够继续进行而不丢失进度，或者如何进行节点恢复，是一个重要问题。

## 2021年

### 1.如何理解分布式系统的透明性？

答：包含多个方面：

- 访问透明性：对不同数据的表示形式以及资源访问的隐藏
- 位置透明性：用户无法感知到资源在系统中的物理位置
- 迁移透明性：分布式系统的资源迁移不会影响到该资源的访问方式
- 重定向透明性：资源可以在接收访问的同时进行重新定位而不引起用户和应用程序的注意
- 复制透明性：对同一资源存在多个副本这样一个事实的隐瞒
- 并发透明性：确保对共享资源的并发访问不会破坏资源的一致状态
- 故障透明性：用户不会注意到某个资源无法正常工作

### 2.持久化通信和非持久化通信的区别

答：瞬时通信：通讯系统只有在发送和接收应用程序正在运行时才能存储信息

持久通信：提交传输的消息一直由通信中间件存储，直到该消息被传送给接收方为止

### 3.RPC中处理客户端崩溃的方法有哪些？

2024年这部分其实考查的是RPC常见的故障情况有哪些，分5种故障类型：

- 客户端不能定位服务器
- 从客户端到服务器的请求消息丢失
- 从服务器到客户端的响应消息丢失
- 服务器在接收消息后故障
- 客户端在发送请求后故障

其中最重要的是后两条，经常考该如何应对这两种故障。

- 答：
- 1.Extermination（消灭）：在客户端存根发送RPC前，构建日志条目明确其目的；重启后日志被检查，孤儿被消灭；
  - 2.Reincarnation（再生）：客户端恢复后广播重启新一个轮次的请求，所有与那个客户有关的远程计算都被杀死；
  - 3.软再生：在新时期广播到达时，每台机器都进行检查查看是否存在远程计算，如果有就尝试定位它的拥有者，只有当无法找到时才杀死该计算；
  - 4.到期：每个RPC都有一个标准时间T，如果到时间还不能结束，就必须显式的请求另一个T。

## 4.同步通信和异步通信含义，TCP是同步还是异步？

答：同步通信是指通信双方在通信过程中必须按照预定的顺序进行操作，发送方等待接收方的响应才能继续执行。

异步通信是指通信双方的操作互不阻塞，发送方在发送请求后不需要等待接收方的响应，而是可以继续执行其他操作。

TCP是同步，当应用程序使用TCP发送数据时，发送方会将数据分割成一个个数据包并发送出去。应用程序通常会通过调用阻塞式的 `send()` 方法进行数据传输，并等待内核层完成数据发送。接收方在收到数据后，会按照顺序进行处理，通常也会通过阻塞式的 `recv()` 方法来接收数据，直到有数据到达。

## 5.结构化P2P和非结构化P2P的区别是什么，分别有什么优劣？

答：**结构化P2P**：节点根据特定的分布式数据结构（逻辑环、超立方）来组织，通过节点ID来响应服务。

- 优点：能实现高效查找，容错性和可靠性高，可扩展性好
- 缺点：管理和维护较为复杂，灵活性差

**非结构化P2P**：节点随机选择邻居

- 优点：灵活性高，实现简单
- 缺点、查找效率低、缺乏全局一致性

## 6.流媒体传输的QOS指标有哪些？

- 答：
- 1.数据传输所要求的比特率
  - 2.创建会话的最大延时
  - 3.端到端的最大延时
  - 4.最大延时抖动
  - 5.最大往返延时

## 7.为什么分布式系统使用逻辑时钟而不使用物理时钟？

答：逻辑时钟关注的是事件发生的顺序和因果关系，不依赖于系统的物理时间，因此在分布式系统中使用更加灵活和可靠。它解决了物理时钟无法同步的问题，确保了事件的相对顺序。物理时钟只能提供节点本地的时间，并且受限于时钟漂移和网络延迟的影响，无法提供可靠的事件顺序信息。

## 2020年

### 1.网络操作系统和分布式系统（基于中间件的）的定义分别是什么？通过实例说明。

答：网络操作系统（NOS）是指通过网络连接多个计算机系统，提供统一的资源管理和协同工作的操作系统。分布式系统是一种由多个独立计算机组成的系统，这些计算机通过网络进行通信和协作，协同完成一些任务。

### 2.为什么Lamport不能全序逻辑但向量钟可以？

答：Lamport 时钟只能提供 偏序关系，不能实现全序，是因为它仅依赖单一的时间戳，无法区分发生在不同节点之间的并发事件。

向量时钟可以提供更精确的时间关系，能够区分并发事件，从而在需要时建立全序。

### 3.分别描述流媒体的非交叉传输和交叉传输的优缺点。

答：非交叉传输：

- 优点：实现简单、低处理开销、适合连续播放
- 缺点：抗丢包能力差、不适合高延迟网络

交叉传输：

- 优点：抗丢包能力强、容错性好、适合实时要求高的场景
- 缺点：实现复杂、处理开销大

## 2019年

### 1.分布式系统的开放性主要体现在哪些方面？

答： 1.提供的组件可以很容易地被其他系统使用或集成

2.无论处于哪种底层环境，分布式系统能够与其他开放系统服务进行交互

- 系统具有良好的接口规范

- 系统支持应用的可移植性
- 系统很容易进行交互

3.至少保证分布式系统独立于底层环境异质性

## **2.在不稳定的广域网上，不改变物理环境的前提下，请给出两个以上的方法来提升在线直播系统的流畅性，并说明这些方法的基本工作原理。**

答： 1.利用P2P技术优化直播流分发：能减轻中心化服务器的负载、提升网络资源利用率  
2.使用交叉传输：抗丢包能力强、容错性好、适合实时要求高的场景

## **3.简述进程和线程的异同，一个web应用采用多线程结构有一定有优势，列举两个。**

答：相同点：

- 都是程序的基本执行单元，用于完成任务
- 都可以并发执行，提高程序效率
- 都依赖操作系统的调度机制分配CPU时间切片

不同点：

- 进程之间相互独立不能共享资源，同一进程内的线程共享内存、文件资源
- 进程之间隔离性强互不干扰；线程隔离性弱，一个线程崩溃可能导致整个进程崩溃

Web浏览器采用多线程分别进行数据分发、图像显示等功能，可以不必等待即可渲染页面

Web服务器使用多线程接收客户端请求，能缓解并发访问的压力，极大减少客户端响应时间

## **4.以数据为中心的一致性和以客户为中心的一致性有什么异同？若要通过增加副本来提升门户网站的性能，副本应该采用哪种类型的一致性？**

答： 以数据为中心的一致性：读写并重（计算密集型任务）。是一种分布式数据仓模型，在物理上分布式共享内存、数据库和文件，在操作上每个进程可执行读写操作，其中写操作在本机副本上执行并广播给其他副本；

以客户为中心的一致性：读多写少（内容分发、手机应用）。一般针对无写写冲突或冲突易解决的分布式系统，保证最终一致性。

门户网站是读多写少，采用以客户为中心的一致性模型即可。

# 2015年

## 1.如何理解分布式系统中的机制和策略？

策略和机制考点在考查时一般是会给出一个例子，然后问你这是策略还是机制，并给出对应的机制或策略。这要去理解一下什么是策略什么是机制，总结来说：

- 策略指做什么，决定系统应采取什么样的行动，不关心具体的实现细节
- 机制指如何做、提供实现策略的具体方法

建议直接做几道题理解一下

答：在开放的分布式系统中要获得灵活性，就要把系统组织成规模相对较小且容易修改和替换的组件，需要将策略和机制分离。

## 2.为什么要采用动态绑定机制来实现服务器定位，动态绑定有什么缺点？

答：将服务器地址烧入客户端不灵活，且动态绑定机制支持多服务器的动态加入退出和不同版本的维护。

缺点是和binder的交互耗时，并且binder很容易造成单点故障。

## 3.为什么要进行同步，分布式系统中的同步和集中式系统中的同步有何区别？

答：很多程序需要依赖于程序创建和修改的先后顺序进行处理，所以在不同的系统或是进程间需要确定先后关系（时间同步）。

集中式中所有时间获取都是向同一个模块获取，相对容易，但分布式系统需要和不同的机器进行交互以达到同步。