

# 分布式2019

---

(基本是给的答案)

---

## 分布式系统的定义与架构

---

- 分布式系统的开放性主要体现在哪些方面？
  - 能够与其他开放系统进行交互，不管底层的环境是什么
  - 系统应该有定义良好的接口
  - 系统应该支持应用程序的可移植性
  - 系统应该容易交互
  - 至少使得分布式系统独立于底层环境的异构性
- 比特流BitTorrent是一种内容分发协议。简要描述基于该协议的分布式内容共享系统的基本架构和工作方式。
  - 混合式架构
  - 集中式的文件目录服务，P2P的文件传输
- 判定以下三项是属于“策略”还是“机制”。如果是策略，写出相应的机制，如果是机制，任意给出一个相应的策略。
  - 模拟应用需求动态设定缓存的大小
    - 策略
    - 对应机制：可以改变系统缓存的大小
  - 支持不同类型的加密系统
    - 机制
    - 对应策略：根据安全性要求，选择一种加密系统
  - 能够通过多条路径传递数据包
    - 机制
    - 对应策略：根据拥塞程度选择一条最合适的路径

## 通信机制

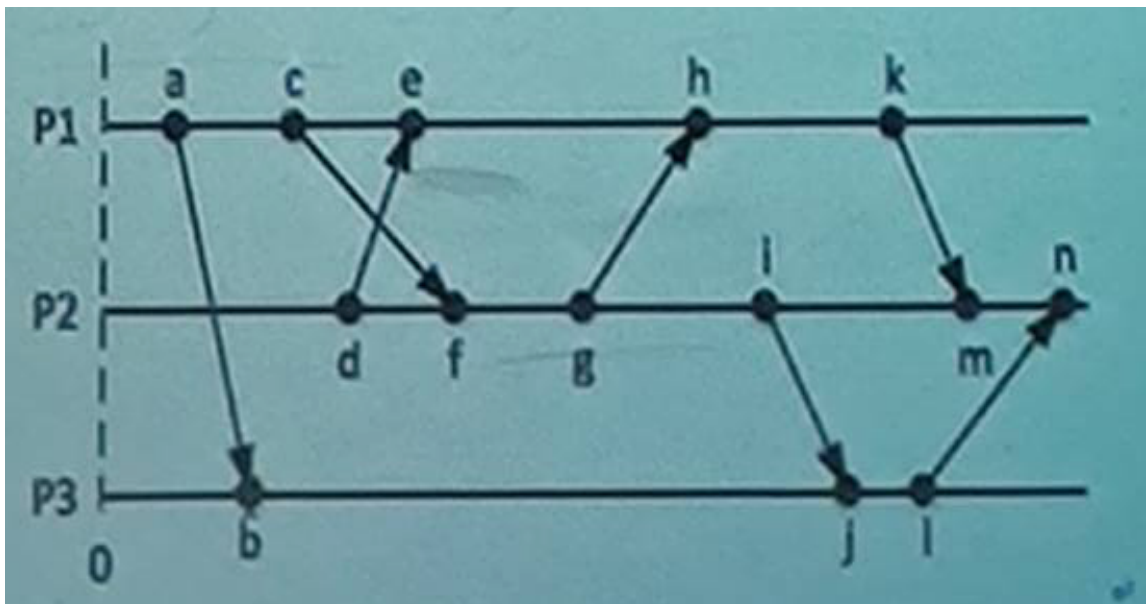
---

- 简述如何在TCP的基础上，实现远程过程调用的“最多执行一次”的机制。（不确定）
  - 在TCP消息中附加一个完整的或者足够长的时间戳历史，以确保能够检测到重复消息，在历史上有时间戳的传入消息被忽略，返回之前的回应而不是重新运行handler，客户端发送多次消息也只会有一次会被执行。
- 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，说明这些方法的基本工作原理。
- 电子邮件采用什么样的通信模式？简要说明如何在一般的TCP/IP协议基础上实现这样的通信。

- 持久性：消息被保存在通信系统中，直到这个通信系统将消息送到接收方，通常称为消息队列系统，因为他在两端都涉及到队列，电子邮件属于持久性通信。
- ？发送者与服务器通过TCP/IP建立连接，将邮件发送到服务器；服务器与接收者通过TCP/IP协议建立连接，将邮件下载到接收者客户端。
- 简述进程与线程的异同，一个Web应用采用多线程结构有一定的优势，列举两个。
  - 见2016
  - Web浏览器使用多线程分别进行数据收发、图像显示、文字显示等操作，可以不必等待即可渲染出页面。
  - Web服务器如tomcat等使用多线程接收客户端请求，缓解并发访问的压力，能极大减少客户端响应时间。

## 同步

- 为什么在分布式系统中不适合用物理时钟进行同步，简述逻辑时钟的基本含义。
  - 在分布式系统中的不同节点间保持它们的时钟一致是一件不容易的事情。因为每个节点的CPU都有自己的计时器，而不同计时器之间会产生时间偏移，最终导致不同节点上面的时间不一致。即使通过算法进行同步，总会有误差，在某些场景下是不能接受的。
  - 逻辑时钟就是为了解决分布式系统中的时序问题，即如何定义a在b之前发生。逻辑时钟是（松耦合）分布式系统的特性，要求的是系统节点进展之间的相对一致性（同步）。只有相关的系统（进程）才需要有逻辑时钟同步，同步的目的是维持事件的顺序性。除时间的基本特性（一维）外，逻辑时钟与标准时钟（物理时钟）之间没有通用意义上的明确的关系。
- 如下图所示，有P1，P2，P3三个进程以及发生的时间a，b，c.....，假定逻辑时钟的初始值为0。
  -



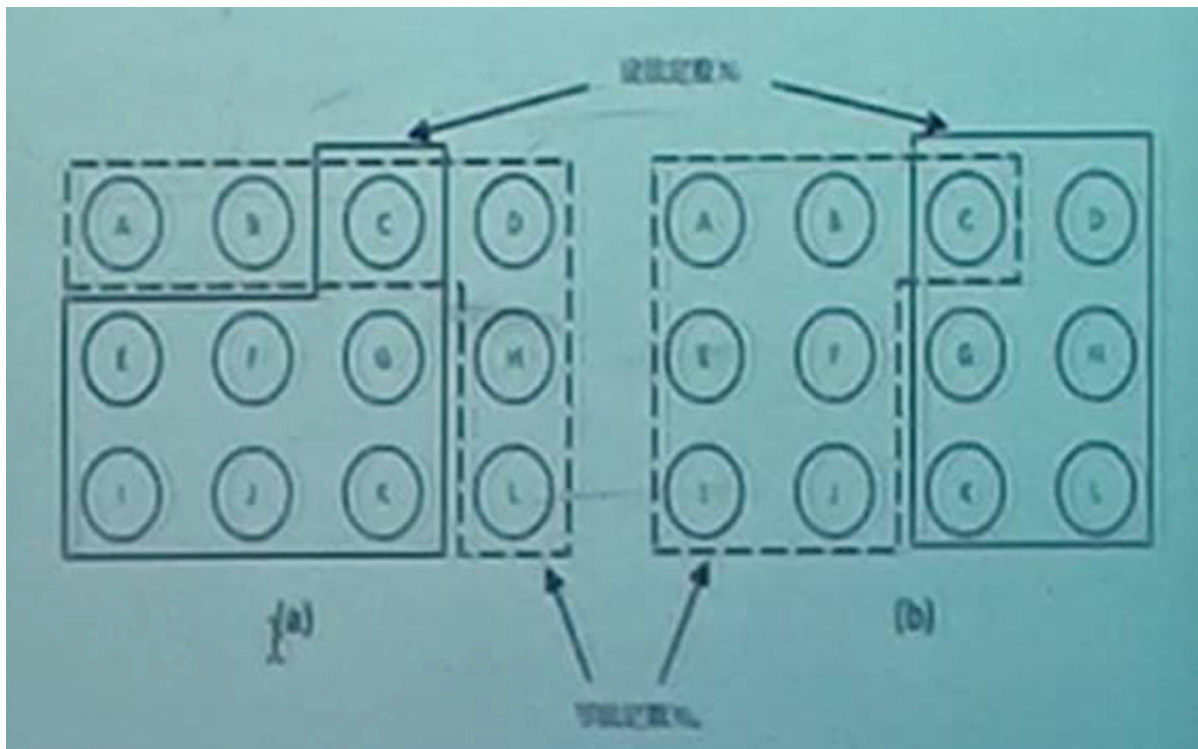
- 为每个时间标定Lamport时间戳，直接在图中标记
- 写出下面六组事件的先后顺序，如a发生在b前写作 $a \rightarrow b$ ，若无法判定打✗
  - (a,f)
  - (d,h)
  - (c,i)

▪ (j,n)

- 设计任一时间戳格式，为每个时间戳定全序逻辑时钟。简要说明方法，并在图中标注。

## 4.一致性

- 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？若希望通过增加副本来提升某门户网站的性能，副本可以采用哪种类型的一致性，给出具体的设计方案。
  - 以数据为中心的一致性模型：读写并重（计算密集型任务）。是一种分布式数据仓库模型，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模型：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，其中大多数操作为读操作，保证了最终一致性，如果很长时间不发生更新操作则所有副本最终一致。
  - 门户网站是读多写少型，选择客户为中心一致性模型。
- 如图a和图b，是否符合法定数量 Quorum-based协议的要求？为什么？



## 5.容错性

- 某个没有协调者的非集中式系统中，正确运行的进程必然返回唯一正确结果，进程出错后会返回任意的错误结果。请设计一个协议保证该系统达到K容错，也就是当K个进程出错时，进程间仍能达成一致并返回正确结果。（需要 $2K+1$ 个进程）
  - 为了取得k故障容错，那么至少需要 $2k+1$ 个进程。
- 由多个传感器从不同位置监控一个系统的温度。该系统的最终温度取这些传感器的平均值，在一个统计周期内，正常工作的传感器对外总是发送其唯一测定的唯一正确温度值；而出现故障的传感器对外发送的温度值是不确定的任意值，但我们无法仅仅通过收到的温度值来判断传感器是否故障。
  - 请设计一个协议保证正确的传感器间能协调一致，并得到正确的温度平均值。

- 如果要求在K个传感器出错的情况下，系统仍然能够正常工作，总共需要安放多少个传感器？  
( $3K+1$ )
  - 在具有k个错误进程的系统中，仅当存在 $2k+1$ 个或更多正常进程并且整体上存在 $N \leq 3k+1$ 个进程时才达成协议。换句话说，只有超过三分之二的进程正常工作才能达成协议。（如果小于该值，则可能会因失败的过程而受到欺骗。）

## 云计算与物联网

- 虚拟化技术在云计算中是如何运用的？如何理解不数据与虚拟化的私有云能提高现有系统性能？
  - 云计算的虚拟化技术不同于传统的单一虚拟化，它是涵盖整个IT架构的，包括资源、网络、应用和桌面在内的全系统虚拟化，它的优势在于能够把所有硬件设备、软件应用和数据隔离开来，打破硬件配置、软件部署和数据分布的界限，实现IT架构的动态化，实现资源集中管理，使应用能够动态地使用虚拟资源和物理资源，提高系统适应需求和环境的能力。  
云计算虚拟化技术的应用意义并不仅仅在于提高资源利用率并降低成本，更大的意义是提供强大的计算能力。计算能力对于系统运行效率、精度和可靠性影响很大，而虚拟化技术可以将大量分散的、没有得到充分利用的计算能力，整合到计算高负荷的计算机或服务服务器上，实现全网资源统一调度使用，从而在存储、传输、运算等多个计算方面达到高效。
- 大数据分析系统大多采用集中式的主从(master-slave)架构，每个提交到系统的作业都会生成一个作业管理器来向系统的主节点索取资源。系统资源管理有两种方案: (6分)
  - 一种是基于“拉”的方式(poll-based): 如Apache Hadoop YARN,每个运行着的作业管理器会定期(例如每3秒)向master报告心跳信息(索要资源)，如果有资源的话，主节点向该作业管理器返回资源分配的结果。作业管理器拿到资源后就会在相应的节点运行其任务。
  - 一种是基于“推”的方式(push-based): 如Apache Mesos,主节点发现有空闲资源时，会主动向作业管理器推送资源;若该作业想要这个资源，那么作业管理器就会发送作业到空闲资源上运行;若不想要，则会给主节点发送拒绝消息，主节点再向其他作业管理器推送。
  - 试分析这两种资源管理方案优缺点，并分别列举一个适合的应用场景。
- 在一个RFID系统中，阅读器通过识别其扫描范围内的标签获取感知数据。为了获取大规模的感知数据，现有n个阅读器(n远远小于标签总数)，他们之间的扫描范围可能存在重叠，且少部分标签的位置会动态变化，如何选取最小数目的阅读器识别最大数目的标签?请提供一种集中式的解决方案和一种分布式的解决方案。(提供解决方案的主要步骤，无需具体计算)。
  - 注: 1) 当多个阅读器同时扫描重叠区域的标签，将无法识别标签: 2) 不存在某一个阅读器位于另一个阅读器扫描范围内的情况: 3)系统部署在二维空间中且阅读器的位置是已知的: 4)阅读器可以根据标签响应情况估算标签分布。
  - 答案:选取最小数目的阅读器识别最大数目的标签，可转化为“带权最大独立集”问题。  
集中式方案:。  
1)将二维空间划分为多个子区域。如一个个格状区域:  
2)在每一个子区域中通过枚举求带权最大独立集:  
3)综合子区域的带权最大独立集，得到全局的带权最大独立集。
  - 分布式方案:。  
1)将二维空间划分为多个子区域，如一个个格状区域:  
2)在每一个子区域中，随机选择一个阅读器作为协调者，估算当前子区域的标签分布，并通过标签分布计算子区域中的近似带权最大独立集:  
3)综合子区域的带权最大独立集，得到全局的带权最大独立集。