# 一． 分布式系统特征

**1. 什么叫分布式系统？举例说明现实生活中的分布式系统，阐述系统组成部分与功能特点。**

分布式系统：分布式系统是一个硬件或软件组件分布在不同的网络计算机上，通过消息传递进行通信和协调的系统。

举例 - Web搜索  
  
·底层物理设施，它由超大数目的位于全世界多个数据中心的联网计算机组成；  
  
·分布式文件系统，支持超大文件，并根据搜索和其他应用的使用方式（特别是在文件中以快速而持久的速度读取）进行了深度优化；  
  
·相关的结构化分布式存储系统，它提供对超大数据集的快速访问；  
  
·锁服务，它提供诸如分布式加锁和协定等分布式系统功能；  
  
·编程模式，它支持对底层物理基础设施上的超大并行和分布式计算的管理。

**2. 分布式系统的特征是什么？**

并发性 - 程序通过共享资源并行  
  
缺乏全局时钟 - 分布式系统中的计算机只能通过本地时钟交换消息；不可预测的消息延迟限制了准确性；没有一个全局时钟的概念。  
  
故障独立性 - 计算机中的故障或程序中的异常马上不能被与之通讯的其他组件感知；系统设计者需要为故障处理做计划。

**3. 给出能被共享的5种类型的硬件资源和5种类型的数据或软件资源。给出它们在实际的分布式系统中发生共享的例子。**

硬盘共享，打印机共享

网络上共享打印机，指接收并处理来自一个以上计算机的打印任务的打印机。

**4. 在分布式系统中，常常说，“避免性能瓶颈”。可否举例说明并谈谈你的观点。**

**5. 在故障处理（Failure handling）中，什么叫容错？什么叫冗余？**

冗余：指重复配置系统的一些部件,当系统发生故障时,冗余配置的部件介入并承担故障部件的工作,由此减少系统的故障时间。通常指通过多重备份来增加系统的可靠性。  
  
容错：容错是用冗余的资源使计算机具有容忍故障的能力，即在产生故障的情况下，仍有能力将指定的算法继续完成。  
  
容错技术是指在一定程度上容忍故障的技术，也称为故障掩盖技术（fault masking）。采用容错技术的系统称容错系统。  
  
容错主要依靠冗余设计来实现，它以增加资源的办法换取可靠性。由于资源的不同，冗余技术分为硬件冗余、软件冗余、时间冗余和信息冗余。

**\*** 容错： 容错性是指软件检测应用程序所运行的软件或硬件中发生的错误并从错误中恢复的能力，使系统能够在某些组件发生故障（或其中的一个或多个故障）时继续正常运行。  
**\*** 冗余： 人为增加地重复部分，其目的是用来对原本的单一部分进行备份，以达到增强其安全性的目的。

**6. 现今分布式操作系统的挑战有：Heterogeneity（异构性），Openness（开放性），Security（安全性），Scalability（可伸缩性），Failure handling（故障处理），Concurrency（并发性），Transparency（透明性），等。分别给出挑战的定义，举例与详细分析挑战涉及的关键技术。**

Heterogeneity（异构性）

定义：在一个分布式系统中，组成该系统的软硬件资源、系统内各成员交互所使用的网络、编写应用或系统所使用的编程语言以及开发者的对一项任务的实施方式等都可能是不同的，这就是分布式系统异构型的体现。

而要解决这各方面的异构性从而使得整个分布式系统能够运作良好，并且达到透明性的目标，并不简单。在一个分布式系统中，你所取得的资源有可能是远在地球另一边的服务器所提供的，在这样的情况下，信息的传输势必会经过不同类型的网络，而不同的网络遵循着各自的网络协议，不同协议间如何沟通，以使得信息能够正确的传输，是一个问题。同时，组成分布式系统的各计算机很有可能是不同的，举例而言，就整型数据的存储来说，采用大端方式存储的计算机与采用小端方式存储的计算机，其对于同一个数（回文数字除外）的存储就截然不同，如忽视这种差异，就可能会得到完全相反的结果。此外，不同的计算机上运行的操作系统也可能是不一样的，而不一样的操作系统其对于接口的实现又非常可能是不同的，如UNIX系统中对于消息交换的系统调用就与Windows系统不同。而不同的编程语言又有着对相同数据结构的不同表示方法，正确识别各编程语言下的数据结构，对于各语言彼此交流非常重要。一个大型的分布式系统的开发通常需要多位开发者，不同的开发者不可避免的存在着自己的编码习惯或者说编码风格，若没有一个统一的开发规范，最终形成的系统很可能不能够成为一个整体运作。

综上，对于任何一个分布式系统，异构性的问题都是亟需解决的一大难点。 Concurrency（并发性）

定义：在一个分布式系统中，存在着若干资源，这些资源可被用户或者说客户端同时访问，如何处理这些访问，以防系统中出现不一致的情况，此即并发性所带来的问题。

举例而言，若12306网站无法很好的处理并发，那么很有可能出现多个用户抢到同一张票的情况，尤其是在春运期间。同理，若微信红包系统无法很好的处理并发，那么很有可能出现多个用户抢到同一个红包的情况。上述的问题都是无法容忍的。而简单的利用加锁操作来限制多个用户同时进行操作，在访问需求非常大的情况下，又会降低吞吐量，使得系统性能下降，用户体验下降，故并非一个很好的解决方案。并发性的解决对于系统的稳定高效运行至关重要。

# 二． 系统模型

**1. 分布式系统模型设计时，设计者常常面对的分布式系统的困难与威胁问题有哪些？**

1)使用模式的多样性，

系统组件承受各种工作负载，如Web每天有几百万的访问量

系统断线或者链接不稳定，如系统中包括移动计算机

系统对宽带与延迟的特殊要求，例如多媒体应用

2）使用模式的多样性：

 系统组件承受各种工作负载，例如：Web每天有几百万的访问量

 系统断线或连接不稳定，例如：系统中包括移动计算机

系统对带宽与延迟的特殊要求，例如：多媒体应用

3）系统环境的多样性：

 容纳异构硬件、操作系统和网络

 网络在性能上有很大的不同，例如：无线网的速度只达到局域网的几分之一

 支持不同规模的系统，从几十台计算机到几百万台计算机

4）内部问题：

 非同步时钟

 冲突的数据更新

 系统组件的软硬件故障

例如：并发引起的数据不一致问题，火车订票问题

5）外部问题：

 数据完整性

 保密性的攻击

 服务拒绝攻击

例如：仿制银行的钓鱼网站等

**2. 分布式系统体系结构元素包括：通信实体、通信范型、角色和责任、放置，以客户-服务器系统体系结构为例解释相应元素概念。**

**3. 什么叫层次化软件体系结构？举例说明。**

**4. 什么叫瘦客户？举例说明你的观点。**

瘦客户：是指一个软件层，它支持用户端的计算机上基于窗口的用户界面，而在远程的计算机上执行应用程序。

**5. 举例说明并图示异步分布式系统中，不同网络结点间进程的“事件的实时排序”。（参考图2-13）**

**6. 掌握分布式系统的设计实例，分布式系统的基础模型有哪些？分别可以解决哪些问题？**

# 三． 进程间通信

**1. 线程与进程的区别？**

**线程:** 线程是操作系统能够进行运算调度的最小单位。 它被包含在进程之中，是进程中的实际运作单位。  
**进程:** 进程是具有独立功能的程序关于某个数据集合上的一次运行活动，是系统进行资源分配和调度的独立单位。

**区别:** 两者完成的工作不同，线程用于小任务，而进程用于更多的'重量级'的任务- 应用基本执行。

(1) 地址空间: 进程内的一个执行单元； 进程至少有一个线程； 它们共享进程的地址空间； 而进程有自己独立的地址空间；  
(2) 资源拥有: 进程是资源分配和拥有的单位， 同一个进程内的线程共享进程的资源  
(3) 线程是处理器调度的基本单位， 但进程不是。  
(4) 二者均可并发执行。

**2. 什么是进程、线程的并发控制与调度？什么是多线程并发控制？**

答：进程调度 - 操作系统管理了系统有限资源，当有多个进程（或多个进程发出的请求）要使用这些资源时，因为资源有限性，必须按照一定的原则选择进程（请求）来占用资源，这就是调度。  
  
线程调度 - 指按照特定机制为多个线程分配CPU的使用权。  
  
多线程并发控制 -是指从软件或者硬件上实现多个线程并发执行的技术。具有多线程能力的计算机因有硬件支持而能够在同一时间执行多于一个线程，进而提升整体处理性能。

**3. 什么是进程间通信？什么是消息传递？给出定义并举例说明。**

答：进程间通信 - 指进程间的信息交换，其所交换的信息量，少则只是一个状态或一个数值，多则可能是成千上万个字节。

**4. 什么是客户/服务器编程？如何进行客户端编程设计？如何进行服务器端编程设计？如何设计服务器与客户的通信？给出基于Socket的客户/服务**

**器，面向UDP和TCP网络编程的详细说明，同时给出一个UNIX/Linux的Client/Sever的程序设计算法流程图。**

1、客户端程序实现 put 功能(将一个文件从本地传到文件服务器) ； put[-h hostname] **[-p portname]** local\_filename remote\_filename   
 2、客户端程序实现 get 功能(从文件服务器取一远程文件存到本地客户端)。注意：客户端和文件服务器不在同一台机器上。get [-h hostname] **[-p portname]** remote\_filename local\_filename  
  
解:  
  
**\*\***伪代码实现**\*\***Client  
  
*```  
function put(server\_ip, server\_port, local\_file\_path) {  
 //register socket connection  
 Socket socket = new Socket()  
 socket.connect(server\_ip, server\_port)  
  
 //add socket read data closure  
 socket.read\_data(void(Data readData){  
 if (readData.is\_upload\_ack) {  
 socket.close()  
 }  
 })  
   
 //process upload file  
 File file = new File(local\_file\_path)  
 Data fileData = new Data(file)  
  
 // send file data  
 socket.write(fileData)  
}  
```*Server  
  
*```  
function get(host\_ip, host\_port, file\_storage\_path) {  
 // bind socket  
 Socket socket = new Socket()  
 socket.bind(host\_ip, host\_port)  
  
 // add socket read data closure  
 socket.read\_data(void(Data readData){  
 if (readData.is\_upload\_req) {  
 File file = new File(readData)  
 // save file  
 file.sync(file\_storage\_path)  
 Data upload\_ack\_data = new Data()  
  
 // send ack  
 socket.write(upload\_ack\_data)  
  
 // close  
 socket.close()  
 }  
 })  
  
 // listen  
 while(true) {  
 socket.listen(void(remote\_ip) {  
 socket.accept()  
 })  
 }  
  
}  
```*

# 四． 远程过程调用

**1. 什么叫远程过程调用？举例说明。**

远程过程调用：即RPC(Remote Procedure Call) 指用户可以向调用本地过程一样调用不同地域的不同计算机上的过程，从而使得应用程序设计人员不必设计和开发有关发送和接收信息的实现细节，体现了分布式系统的透明性.  
  
举例：用户使用移动端设备买票，执行抢票动作的机器可能在北京可能在上海，移动设备可以直接通过协议远程调用买票的指令且不用关心远端机房具体的位置。

**2. 描述远程过程调用RPC的工作原理和RPC调用的步骤。**

**远程过程调用步骤**

客户过程以正常的方式调用客户存根

•客户存根生成一个消息，然后调用本地操作系统

•客户端操作系统将消息发送给远程操作系统

•远程操作系统将消息交给服务器存根

•服务器存根将参数提取出来，然后调用服务器

•服务器执行要求的操作，操作完成后将结果返回给服务器存根

•服务器存根将结果打包成一个消息，然后调用本地操作系统

•服务器操作系统将含有结果的消息发送回客户端操作系统

•客户端操作系统将消息交给客户存根

•客户存根将结果从消息中提取出来，返回给调用它的客户过程

**3. 描述远程过程调用通信模型。**

**4. 使用socket编程实现一个简单的远程过程调用流程设计，文件系统存放在远端服务器上，要求：**

1）客户端程序实现put功能(将一个文件从本地传到文件服务器) ；put [-h hostname] [-p portname] local\_filename remote\_filename

2）客户端程序实现get功能(从文件服务器取一远程文件存到本地客户端)。注意：客户端和文件服务器不在同一台机器上。

get [-h hostname] [-p portname] remote\_filename local\_filename

# 五． 操作系统支持

**1. 什么叫虚拟机？什么叫虚拟化？举例说明。**

答：虚拟机指通过软件模拟的具有完整硬件系统功能的、运行在一个完全隔离环境中的完整计算机系统。  
  
虚拟化就是将事物从一种形式转变成另一种形式，最常用的虚拟化技术有操作系统中内存的虚拟化，实际运行时用户需要的内存空间可能远远大于物理机器的内存大小，利用内存的虚拟化技术，用户可以将一部分硬盘虚拟化为内存，而这对用户是透明的。又如，可以利用虚拟专用网技术（VPN）在公共网络中虚拟化一条安全，稳定的“隧道”，用户感觉像是使用私有网络一样。

**2. 什么叫保护？举例说明。**

**3. 什么叫进程调度？什么叫线程调度？多线程并发处理？**

答：进程调度 - 操作系统管理了系统有限资源，当有多个进程（或多个进程发出的请求）要使用这些资源时，因为资源有限性，必须按照一定的原则选择进程（请求）来占用资源，这就是调度。  
  
线程调度 - 指按照特定机制为多个线程分配CPU的使用权。  
  
多线程并发控制 -是指从软件或者硬件上实现多个线程并发执行的技术。具有多线程能力的计算机因有硬件支持而能够在同一时间执行多于一个线程，进而提升整体处理性能。

**4. 什么叫临界区？什么叫临界资源？如何解释线程同步？**

临界区 - 指的是一个访问共用资源（例如：共用设备或是共用存储器）的程序片段，而这些共用资源又无法同时被多个线程访问的特性。当有线程进入临界区段时，其他线程或是进程必须等待（例如：bounded waiting 等待法），有一些同步的机制必须在临界区段的进入点与离开点实现，以确保这些共用资源是被互斥获得使用。  
  
临界资源 - 一次仅允许一个进程使用的资源称为临界资源。  
  
线程同步 - 在一般情况下，创建一个线程是不能提高程序的执行效率的，所以要创建多个线程。但是多个线程同时运行的时候可能调用线程函数，在多个线程同时对同一个内存地址进行写入，由于CPU时间调度上的问题，写入数据会被多次的覆盖，所以就要使线程同步。线程同步即当有一个线程在对内存进行操作时，其他线程都不可以对这个内存地址进行操作，直到该线程完成操作， 其他线程才能对该内存地址进行操作，而其他线程又处于等待状态，目前实现线程同步的方法有很多，临界区对象就是其中一种。  
  
·线程同步就是线程排队。同步就是排队。线程同步的目的就是避免线程“同步”执行。  
  
·只有共享资源的读写访问才需要同步。如果不是共享资源，那么就根本没有同步的必要。   
  
·只有“变量”才需要同步访问。如果共享的资源是固定不变的，那么就相当于“常量”，线程同时读取常量也不需要同步。  
  
·多个线程访问共享资源的代码有可能是同一份代码，也有可能是不同的代码；论是否执行同一份代码，只要这些线程的代码访问同一份可变的共享资源，这些线程之间就需要同步。

**5. 单处理机进程调度算法有哪些？如何描述？**

**6. 影响远程调用的哪些因素会影响消息传递？**

**7. 举例说明在分布式环境下，什么是代码迁移？**

定义：将程序(或执行中的程序)传递到其它计算机。  
  
举例：如大型电商活动中，远端机器A的负责计算优惠信息的进程 被机器B(负责展示)和机器C(负责交易)频繁访问，则可以将计算优惠信息的进程通过代码迁移，迁移到机器B和C下进行，减小机器A的压力，同时提高B和C的响应速度

**8. 举例说明在分布式环境下的处理器任务分配。比较与单处理机分配的不同？**

# 六． 同步化

1. 什么是物理时钟的同步算法？

2. 什么是基于逻辑时钟的同步算法？

3. 如何进行一致性全局状态的检测？

4. 什么是选举算法？

5. 什么是互斥算法？

6. 如何进行分布式系统的死锁处理？

# 七． 分布式文件系统\*(选做——思考题)

1. 分布式文件系统的特点与需求是什么？

2. Sun网络文件系统（NFS）的自动安装器是如何改进NFS的性能和可伸缩性的？

八． 分布式系统前沿与案例分析（**+** 聂慧静老师）

1. 什么是大数据？什么是云平台？举例说明它们的关系。

2. 什么是网格？网格平台与云平台的联系与区别？

3. 案例分析题，以保险公司案例为背景回答下列问题：

1) 传统BI的关键技术包括哪些，分别详细描述。

2) 描述一下保险行业大数据四层架构，大数据在保险行业的应用场景有哪些，选择一个案例详细描述。

3) 描述云服务架构的三个层次分别是什么？保险企业一般重点开发建设的是哪个层次的云架构？

4) 分布式系统中，手机端应用案例的后台架构图中，有哪些关键模块，分别的作用是什么？