**一、 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**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。

**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。

**二、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。

**三、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**。**给出共享的例子。**

硬件资源: 硬盘、 cpu、 gpu、 内存、 打印机

数据或软件资源： 视频、 图片 、文档、 可执行程序、 html

举例：打印机共享，接受并处理来自一个以上计算机的打印任务。通过将打印机物料地连接到服务器上，并在打印机服务器上安装合适的打印机驱动程序，然后将打印机在网络上共享，共享依赖于NOS。

**四、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。**

由于服务和数据分布在不同的机器上，每次交互都需要跨机器运行，会导致网络延迟和网络故障，使系统整体性能降低，会带来一系列的问题，比如资源的锁住，所以系统调用一般都要设置一个超时时间进行自我保护，但是过度的延迟就会带来系统的RPC调用超时，引发一个令人头疼的问题：分布式系统调用的三态结果：成功、失败、超时。不要小看这个第三态，这几乎是所有分布式系统复杂性的根源。要避免这样的性能瓶颈，有一些相应的解决方案：异步化，失败重试。 而对于跨IDC数据分布带来的巨大网络因素影响，则一般会采用数据同步，代理专线等处理方式。

**五、 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。

**六、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。

百度：

一：异构性。网络、计算机硬件、操作系统、编程语言、由不同的开发者完成的软件实现都是造成异构性的主因。其中网络的异构通过互联网协议相互通信而被屏蔽；中间件的流行屏蔽的底层网络、硬件、操作系统和编程语言的异构，它为分布式应用和服务器提供了一直的计算模型，包括RPC、远程事件通知、远程SQL访问和分布式事物调用；虚拟机也是使代码到处运行的一种方法。

二：开放性。它取决于新的资源共享服务能被增加和供多种客户程序使用的程度。特征就是发布系统的关键接口，使其基于一致的通信机制，让不同提供商提供异构硬件和软件。

三：安全性。包括三个部分：机密性（防止泄露给未授权的个人）、完整性（防止被改变或被破坏）、可用性（防止对访问资源的手段的干扰）。其中有两个重要的安全问题：拒绝服务攻击和移动代码的安全性仍然没有得到圆满解决。

四：可伸缩性。顾名思义就是随着资源数量的增加和用户访问的增加，系统仍然能保持其有效性，该系统就被称为可伸缩的。其中有控制物理资源的开销、控制性能损失、防止软件资源用尽和避免性能瓶颈四大挑战。

五：故障处理。在一个分布式系统中，硬件或软件都会出现未知的故障或者不正常运行，因此故障处理是贯穿整个系统的难题。容错（设计容错机制如重传）、故障恢复（数据恢复或“回滚”保证一致性）、冗余（多条路由或者备份等技术）都是故障处理技术。

六：并发性。多个用户对同一资源的使用，要保持操作的正确性就必须在数据保持一致的基础上同步。如使用操作系统的信号量。

七：透明性，指本来是存在的事物或属性，但从某个角度看似乎不存在的现象。系统透明性大致包含：1. 访问透明性：用相同的操作访问本地资源和远程资源（电子邮件）

2. 位置透明性：不需要知道资源的物理位置或网络位置（电子邮件、URL）

3. 并发透明性：几个进程能并发的使用共享资源而不互相干扰

4. 复制透明性：使用资源的多个实例提升可靠性和性能，而用户和程序员无需知 道副本的相关信息

5. 故障透明性：屏蔽错误

6. 移动透明性：资源和客户能够在系统内移动而不受影响（移动电话）

7. 性能透明性：负载变化时，系统能够被重新配置以提高性能

8. 伸缩透明性：系统和应用能够进行扩展而不改变系统结构和应用算法

**七、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，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

2）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

3）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

4）外部问题：

数据完整性

保密性的攻击

服务拒绝攻击

**八、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。**

通信实体 - 客户端与服务器端的通信（对象，组件，web服务），底层进程间的通信。

通信范型 - 分布式系统中实体如何通信。客户-服务器结构中最常见的通信范型是远程调用，如用于支持客户-服务器计算的请求-应答协议是一个有效的模式。

角色和责任 - 在客户-服务器结构中，进程扮演服务器和客户的角色。特别是，为了访问服务器管理的共享资源，客户进程可以与不同主机上的服务器进行交互。

放置 - 对象或服务这样的实体是怎样映射到底层的物理分布式基础设施上的，物理分布式基础设施由大量的机器组成，这些机器通过一个任意复杂的网络互联。从决定分布式系统特性的角度而言，放置是关键的，这些特性大多数与性能相关，也包括其他特性如可靠性和安全性。放置需要考虑实体间的通信模式、给定机器的可靠性和它们当前的负载、不同机器之间的通信质量等。

**九、什么叫层次化软件体系结构？举例说明。**

层次化体系结构与分层体系结构是互补的，分层将服务垂直组织成抽象层，而层次化是一项组织给定层功能的技术，它把这个功能放在合适的服务器上，或者作为第二选择放在物理节点上。

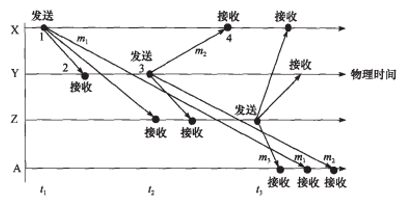
三层层次体系结构: 显示层、应用层、永久数据存储层

**十、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。尽可能少地进行本地数据处理，大部分数据与服务器通信中获得。

例如，智能电话和其他资源有限的设备。缺点：在交互频繁的图形活动中，因为网络和操作系统延迟，而变得不可接受。

**十一、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）



逻辑上，我们知道消息在发送之后才能被接收，因此我们给出一个逻辑排序仅涉及到X和Y：X在接收到m1前发送m1；Y在X接收到m2前发送m2，我们知道应答在接收到消息后发出，因此对于Y，我们有下列逻辑排序：Y在发送m2前接收m1,。逻辑时间通过给每个事件赋予一个与它逻辑相对应的数字而进一步拓展了这个思想，这样，后发生的事件的数字比早发送的事件的数字大，如上图。

**十二、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？

1)交互模型 - 交互模型处理分布式系统中性能以及设置时间限制的困难，例如对于消息传递。反映了进程交互的方式；

2)故障模型 - 故障模型试图给出由进程和通信通道呈现出来的故障的一个精确的规格说明。它定义可靠的通信和正确的进程；

3)安全模型 - 安全模型讨论对于进程和通信通道可能存在的威胁，它引入了安全通道的概念，以低于这些威胁。

**十三、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。

**十四、 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？**

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。

**十五、什么是进程间通信？什么是消息传递？给出定义并举例说明。**

**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。

**管道通信：**即发送进程以字符流形式将大量数据送入管道，接收进程可从管道接收数据，二者利用管道进行通信。

**消息队列：**是在消息的传输过程中保存消息的容器。

信号量机制：即利用PV操作来对信号量进行处理。

**共享内存：**在多处理器的计算机系统中，可以被不同中央处理器（CPU）访问的大容量内存。

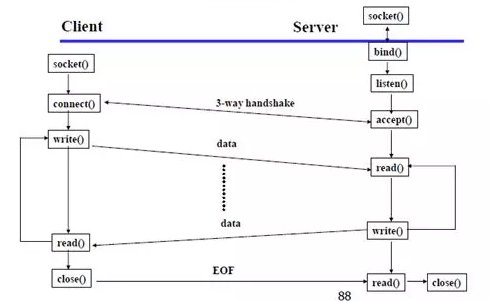
**远程消息传递**：是一种通信的形式，在并发计算、并行计算、面向对象编程与进程间通信中使用。在这种模式中，进程或对象以发送及接收消息的方式来达成同步。不同于传统程序设计通过名字直接调用（invoking）一个进程、子例程或者函数，消息传递直接发送消息给一个进程，依赖进程或基础框架来调用实际执行的代码。可分为同步方式与异步方式。对于分布式对象，发送方和接收方可以存在于不同的计算机上，运行不同的操作系统，使用不同的编程语言等。在这种情况下，总线层负责处理从一个系统到另一个系统的数据转换的详细信息，通过网络等，这就是远程消息传递。

**十六、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/服务器编程：**

**服务器端：**有一个进程（或多个进程）在指定的端口等待客户来连接，服务程序等待客户的连接信息，一旦连接上后，就可以按照设计的数据交换方法和格式进行数据传输。

**客户端：**在需要的时刻发出向服务端的连接请求。



**2）客户端设计（TCP）**

①使用socket调用得到一个socket描述符②使用connect向指定的服务器上的指定端口发起连接，一旦连接成功返回，就说明已经建立了与服务器的一个连接。③通过socket描述符进行读写操作。

**3）服务端设计（TCP）：**

首先启动服务器，过一些时间启动客户端，并连接到服务器

①使用socket调用得到一个描述符②使用bind调用一个名字与socket描述符连接起来，对Internet域就是将Internet地址连编导socket③服务端使用listen调用指出等待请求队列的长度④使用accept调用等待客户端发起连接，一旦有客户端发起连接，accept返回客户独爱的地址信息，并返回一个新的socket描述符，该描述符与原先的socket有相同的特征⑤服务端使用心得socket进行读写操作，说明一般服务端可能在accept返回后创建一个新的进程进行与客户端的通信，父进程则再到accept调用处等待另一个连接

**4）无连接协议UDP**

①客户不是可服务器建立连接，而是使用sendto系统调用向服务器发一个数据包，作为参数请求的地址（服务端）②服务器也不必接收来自客户短的连接，而是发送系统调用recvfrom，等待来自客户端的数据到达，此系统调用返回客户端进程的网络地址和数据包服务器可向对方进程发出相应。

四． 远程过程调用

**十七、什么叫远程过程调用？举例说明。**

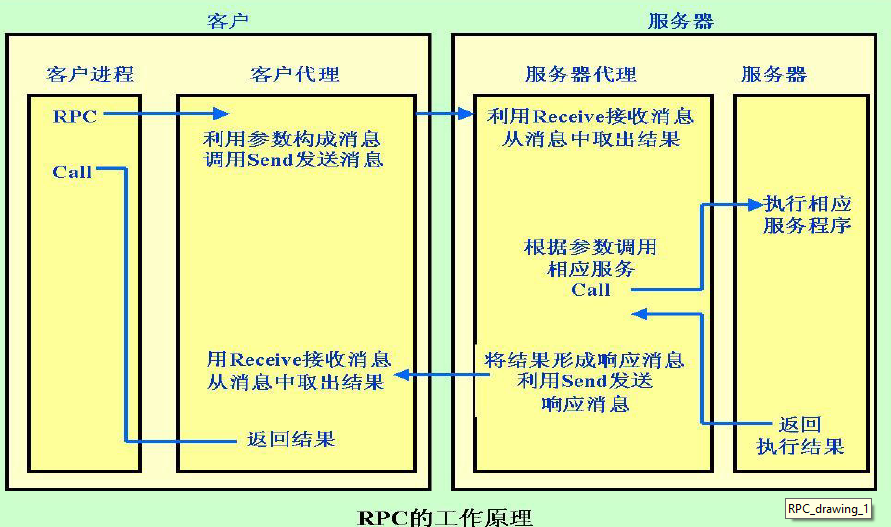
**远程过程调用**：即RPC(Remote Procedure Call) 指用户可以向调用本地过程一样调用不同地域的不同计算机上的过程，从而使得应用程序设计人员不必设计和开发有关发送和接收信息的实现细节，体现了分布式系统的透明性.

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。

**十八、影响远程调用的哪些因素会影咱消息传递**

1）传递参数的方式：客户机与服务器的机器类型不同，数据表示不同。2）客户如何定位服务器，采用动态联编。3）出现差错时的RPC语义：客户找不到服务器，请求或应答消息丢失，接受或发送消息后突然崩溃

**十九、描述远程过程调用RPC的工作原理和RPC调用的步骤。**



RPC采用客户机/服务器模式。请求程序就是一个客户机，而服务提供程序就是一个服务器。首先，客户机调用进程发送一个有进程参数的调用信息到服务进程，然后等待应答信息。在服务器端，进程保持睡眠状态直到调用信息到达为止。当一个调用信息到达，服务器获得进程参数，计算结果，发送答复信息，然后等待下一个调用信息，最后，客户端调用进程接收答复信息，获得进程结果，然后调用执行继续进行。

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

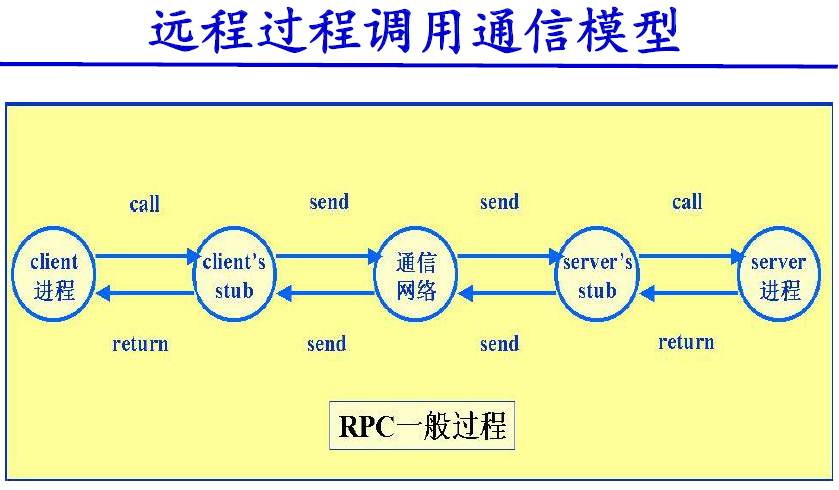
•服务器存根将结果打包成一个消息，然后调用本地操作系统

•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

**二十、描述远程过程调用通信模型。**



**二十一、 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

【Client】

public void 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】

public void 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()

})

}

}

**二十二、分布式负载均衡**

广义上，负载均衡就是把数据从一个机器分发发到多个机器进行处理。负载均衡建立在现有网络结构之上，它提供了一种廉价有效透明的方法扩展网络设备和服务器的带宽、增加吞吐量、加强网络数据处理能力、提高网络的灵活性和可用性。

负载均衡（LoadBalance）其意思就是分摊到多个操作单元上进行执行，例如Web服务器、FTP服务器、企业关键应用服务器和其它关键任务服务器等，从而共同完成工作任务。

**二十三、 什么叫虚拟机？什么叫虚拟化？举例说明。**

**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。

**虚拟存储**（StorageVirtualization）是指将多个不同类型、独立存在的物理存储体，通过软、硬件技术，集成转化为一个逻辑上的虚拟的存储单元，集中管理供用户统一使用。这个虚拟逻辑存储单元的存储容量是它所集中管理的各物理存储体的存储量的总和，而它具有的访问带宽则在一定程度上接近各个物理存储体的访问带宽之和。

**二十四、什么叫保护？举例说明。**

**保护：**保护是指一种控制程序、进程或用户对计算机系统资源进行访问的机制。操作系统中的进程必须加以保护，使其免受其他进程活动的干扰。为此，系统采用了各种机制确保只有从操作系统中获得了恰当授权的进程才可以操作相应的文件、内存段、CPU和其他的资源。

**举例：**为了防止文件共享可能会导致文件被破坏或未经核准的用户修改文件，文件系统必须控制用户对文件的存取，即解决对文件的读、写、执行的许可问题。为此，必须在文件系统中建立相应的文件保护机制。

文件保护通过口令保护、加密保护和访问控制等方式实现。其中，口令保护和加密保护是为了防止用户文件被他人存取或窃取，而访问控制则用于控制用户对文件的访问方式。

**二十五、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。

**二十六、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**：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。

**二十七、单处理机进程调度算法有哪些？如何描述？**

1)先来先服务调度算法：每次调度都是从队列中选择一个或多个最先进入该队列的进程

2)短作业(进程)优先调度算法 ：SPN 算法从就绪队列中选出估计运行时间最短的进程，为之分配处理机

3）高响应比优先调度算法：为了防止长作业（进程）被饿死，选择最高相应比优先算法。

4）时间片轮转法：就绪队列中的所有进程在一给定的时间内均能获得一时间片的处理机执行时间

5）多级反馈队列调度算法 ：根据进程属性设置多个就绪队列，每个就绪队列有各自的调度算法、优先级或时间片长度等进行多级顺序调度。

**二十八、影响远程调用的哪些因素会影响消息传递？**

1）传递参数的方式：客户机与服务器的机器类型不同，数据表示不同。

2）客户如何定位服务器，采用动态联编。

3）出现差错时的RPC语义：客户找不到服务器，请求或应答消息丢失，接受或发送消息后突然崩溃

**二十九、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。

进程迁移：就是将一个进程的状态，从一台机器(源机)转移到另一台机器(目标机)上，从而使该进程能在目标机上执行。

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

**三十、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？**

图论确定算法、集中式算法、层次性算法、超载者启动的分布式启发算法、欠载者启动的分布式启发算法、拍卖算法

与单处理及不同的是分布式算法还需要考虑每台机器负载的度量，收集负载信息以及传送进程的额外开销，以及其复杂性和稳定性。

**三十一、分布式同步化**主要用来解决分布式环境当中多个进程之间的同步控制，让他们有序的去访问某种临界资源，防止造成"脏数据"的后果。

**三十二、什么是物理时钟的同步算法？**

物理时钟是真实时间，但计算机时钟与其他时钟一样并不完全一致，两个时钟会因为时钟偏移而读数不同，所有要同步时间就有了物理时钟的同步算法，基于物理时钟的同步算法有 Cristian算法和Berkeley算法。

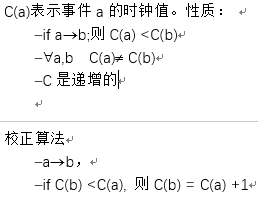
Cristian算法：有一个时间服务器，提供标准时钟，其他系统通过询问与它同步。误差周期内，每个机器向服务器发出校时请求，服务器用自己的时间进行响应，各机器根据响应值重置自己的时钟。

Berkeley算法: 时间服务器没有标准时钟，它通过定期地询问各个机器的当前时间并从中求出平均值作为当前的标准时间，然后再广播给各个机器。当前时钟慢于新标准时间的机器重置自己的时钟，当前时钟快于新标准时间的机器要调整自己的H值以消化这个时间误差，时间服务器的时钟由系统管理员手工校正。

**三十三、什么是基于逻辑时钟的同步算法？**

逻辑时钟是分布式系统的特征，要求的是系统节点进程之间的相对一致性。只有相关的进程才需要有逻辑时钟同步，同步的目的是维持时间的顺序性，除时间的基本特性外，与物理时钟之间没有通用意义上的明确的关系。

Lamport算法，Lamport逻辑时钟是一个单调增长的软件计数器，他的值与任何物理时钟无关，是相对时间。要求系统节点进程之间的相对一致性，只有相关的进程才需要有逻辑时钟同步，同步的目的是维持时间的顺序性。



**三十四、 如何进行一致性全局状态的检测？**

在分布式系统中，按照一致割（一致割是指处理器可以并发保留的状态）来划分进程的时间，可以得到一个系统全局快照。

Chandy-Lamport快照算法：每个进程记录它的状态，对每个接入通道还记录发送给它的消息，对每个通道，进程记录在它自己记录下状态之后和在发送方记录下它自己状态之前达到的任何消息。

**三十五、什么是选举算法？**

选择一个唯一的进程来扮演特定角色的算法称为选举算法(来自书)。

在分布式进程之间做出统一 的决定（确定协调者）（来自课件）。

选举算法是分布式一致性算法用来解决一个分布式系统如何就某个值(决议)达成一致的问题。在一个分布式数据库系统中，如果各节点的初始状态一致，每个节点都执行相同的操作序列，那么他们最后能得到一个一致的状态。为保证每个节点执行相同的命令序列，需要在每一条指令上执行一个“一致性算法”以保证每个节点看到的指令一致。

**三十六、什么是互斥算法？**

答：当一个进程使用某个共享资源，其他进程不允许对这资源操作(来自课件)

当有多个进程竞争系统中相同的资源时，互斥问题就产生了。一个正确的互斥算法必须避免冲突(死锁和饿死) 和保证公平性。通常分布式操作系统使用的互斥算法有:集中式算法、分布式算法和令牌环算法。评价互斥算法性能的准则有(1)每次临界区请求所需要的消息的个数(2)响应时间(RT)(3)同步延迟(SD)。一个好的互斥算法应该是无饥饿,无死锁的。

**三十七、 如何进行分布式系统的死锁处理？**

–鸵鸟法：留给用户考虑

–检测法：发现死锁，进行处理

–预防法：在设计上使死锁不可能发生

–避免法：在运行中，防止出现死锁

**三十八、分布式文件系统的特点与需求是什么？**

特点：可扩展性，高可用性，协议和接口多样性，弹性存储

需求：1）透明性：访问、位置、移动、性能、扩展的透明性2）并发文件更新：并发控制，客户改变文件的操作不影响其他用户访问或改变同一文件的操作3）文件复制：多个副本4）硬件和操作系统异构性：文件服务的接口必须有明确的定义，在不同操作系统和计算机上实现客户和服务器软件5）容错6）一致性7）安全性：身份验证，访问控制，安全通道8）效率：应提供比传统文件系统相同或更强的性能和可靠性

**三十九、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？**

**四十、什么是大数据？什么是云平台？举例说明它们的关系。**

**大数据：**指无法在一定时间范围内用常规软件工具进行捕捉、管理和处理的数据集合，是需要新处理模式才能具有更强的决策力、洞察发现力和流程优化能力的海量、高增长率、高价值和多样化的信息资产。

**特征：体量Volume**

非结构化数据的超大规模和增长，总数据量的80~90%，比结构化数据增长快10倍到50倍，是传统数据仓库的10倍到50倍

**多样性Variety**

大数据的异构和多样性，很多不同形式（文本、图像、视频、机器数据），无模式或者模式不明显，不连贯的语法或句义

**价值密度Value**

大量的不相关信息，对未来趋势与模式的可预测分析，深度复杂分析（机器学习、人工智能Vs传统商务智能，(咨询、报告等）

**速度Velocity**

实时分析而非批量式分析，数据输入、处理与丢弃，立竿见影而非事后见效

**云平台：**指基于硬件的服务，提供计算、网络和存储能力。云平台分三类，以数据存储为主的存储型云平台，以数据处理为主的计算型云平台以及计算和数据存储处理兼顾的综合云计算平台。

**关系：**从技术上看，大数据与云计算的关系就像一枚硬币的正反面一样密不可分。大数据必然无法用单台的计算机进行处理，必须采用分布式架构。它的特色在于对海量数据进行分布式数据挖掘。但它必须依托云计算的分布式处理、分布式数据库和云存储、虚拟化技术。

**四十一、什么是网格？网格平台与云平台的联系与区别？**

答：（来自课件）

网格是信息社会的网络基础设施，它把整个因特网整合成一台巨大的超级虚拟计算机，实现互联网上所有资源的互联互通，完成计算资源、存储资源、通信资源、软件资源、信息资源、知识资源等智能共享的一种新兴的技术。

联系与区别 - 云计算和网格彼此相关，但又有不同。网格计算强调资源共享，任何人都可以做为请求者使用其它节点的资源，任何人都需要贡献一定资源给其他节点。网

格计算强调将工作量转移到远程的可用计算资源上。云计算强调专有，任何人都可以获取自己的专有资源，并且这些资源是由少数团体提供的，使用者不需要贡献自己的资源。在云计算中，计算资源被转换形式去适应工作负载，它支持网格类型应用，也支持非网格环境，比如运行传统或Web2.0应用的三层网络架构。网格计算侧重并行的计算集中性需求，并且难以自动扩展。云计算侧重事务性应用，大量的单独的请求，可以实现自动或半自动的扩展。

**四十二、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？

答：（来自百度）

1）传统BI的关键技术主要包括ETL(数据的提取、转换与加载)技术和数据仓库与数据集市技术、OLAP技术、数据挖掘技术与数据的发布与表示技术。

a. 数据仓库技术：实施BI首先要从企业内部和企业外部不同的数据源，如客户关系管理(CRM)、供应链管理(SCM)、企业资源规划(ERP)系统以及其他应用系统等搜集有用的数据，进行转换和合并，因此需要数据仓库和数据集市技术的支持。

b. OLAP联机分析处理(Online Analytical Processing ，简称OLAP) 又称多维分析，由EF Codd 在1994 年提出，它对数据仓库中的数据进行多维分析和展现，是使分析人员、管理人员或执行人员能够从多种角度对从原始数据中转化出来的、能够真正为用户所理解的、并真实反映企业维特性的信息进行快速、一致、交互地存取，从而获得对数据更深入了解的一类软件技术。它的技术核心是“维”这个概念，因此OLAP也可以说是多维数据分析工具的集合。

c. 数据挖掘技术：与OLAP的探测式数据分析不同，数据挖掘是按照预定的规则对数据库和数据仓库中已有的数据进行信息开采、挖掘和分析，从中识别和抽取隐含的模式和有趣知识，为决策者提供决策依据。数据挖掘的任务是从数据中发现模式。模式有很多种，按功能可分为两大类：预测型( Predictive)模式和描述型(Descriptive)模式。

d. 数据挖掘技术与数据的发布与表示技术：为了使分析后的数据直观、简练地呈现在用户面前，需要采用一定的形式表示和发布出来，通常采用的是一些查询和报表工具。不过，目前越来越多的分析结果是以可视化的形式表现出来，这就需要采用信息可视化技术。所谓信息可视化是指以图形、图像、虚拟现实等易为人们所辨识的方式展现原始数据间的复杂关系、潜在信息以及发展趋势，以便我们能够更好地利用所掌握的信息资源。

2）四层架构：存储层、资源及数据管理层、计算引擎、基于计算引擎的高级封装及工具

3）云服务框架的三个层次：基础设施层、平台层和软件服务层

4）无

**四十三、微服务应用目前成熟的部署方案中，有哪些关键模块，分别的作用是什么？**

微服务应用部署目前主要有负载均衡、 服务注册发现、 智能路由、自动熔断、日志服务、配置中心等关键模块。具体作用如下：

**负载均衡**：实现服务网关的负载均衡，常用解决方案 F5、 Nginx。

**服务注册发现**：服务端通过 REST 协议暴露服务，提供应用服务的注册和发现的功能，常用解决方案 Eureka、 Consul。

**智能路由**：进行安全验证、动态路由、负载分配等，常用解决方案 Zuul。

**自动熔断**：在服务出现超时、异常时，自动熔断，避免服务器瘫痪，常用解决方案：Hystrix。

**统一日志服务**：各应用日志文件采集到统一的服务器上，方便日志查看，常用解决方案： ELK(Elasticsearch、 Logstash、 Kibana)等。

**分布式配置中心**：例如有配置文件发生变化，会通过消息总线统一发布到服务，服务实时更新内存，不需要重启，常用解决方案： Srping Cloud Config、携程 Apollo。

**四十四、传统 BI 的关键技术包括哪些，分别详细描述。**

传统 BI 的关键技术包括 ETL 数据抽取、转换、加载；联机分析处理；数据仓库模型设计；数据的发布（报表开发）等技术。

**• ETL 数据抽取**： 在构造数据仓库时，要经过数据的清洗、数据的抽取转换、数据集成和数据加载等过程。面向不同的需求，对数据进行清洗以保证数据的正确性，然后对数据进行抽取，转换成数据仓库所需形式，并实现加载到数据仓库。

**• 联机分析处理（OLAP）**： 又称多维分析，对数据仓库中的数据进行多维分析和展现，是使分析人员、管理人员或执行人员能够从多种角度对从原始数据中转化出来的真实反映企业维特性的信息进行快速、一致、交互地存取，从而获得对数据更深入了解的一类软件技术。它的技术核心是“维”这个概念，因此 OLAP 也可以说是多维数据分析工具的集合。

**• 数据仓库建模**： 数据仓库的数据模型有星型模式、雪花模式。星型模式最为常见，有一个包含大批数据并且不含冗余的中心表，每维一组小的附属表。 雪花模式中某些维表是规范化的，因而把数据进一步分解到附加的表中，模式图形成了类似

雪花的形状。

**• 数据的发布**： 为了使分析后的数据直观、简练地呈现在用户面前，通常采用的是一些查询和报表工具和信息可视化技术。所谓信息可视化是指以图形、图像、虚拟现实等易为人们所辨识的方式展现原始数据间的复杂关系、潜在信息以及发展趋势，以便我们能够更好地利用所掌握的信息资源。

**四十五、描述一下保险行业大数据四层架构，在保险行业的应用场景有哪些，选择一个案例详细描述实施步骤。**

保险行业大数据四层架构：

L1：基础设施、数据源：关键需要引入保险行业适配的数据，如中保信数据、医疗数据等等。L2：数据平台层：统一的数据集成、存储和分析，统一的数据治理和应用平台L3：数据服务层：整合第三方组件的能力，如：DB、 DWH， Hadoop、 分析工具、 可视化软件L4：数据应用层：保险行业驱动的应用精准营销、智能核保、风险预警等精准营销实施步骤：

1、 数据采集

数据采集分实时采集和批量采集。实时数据采集内容：用户的行为数据，包括设备信息浏览轨迹、页面停留时间、地域等。 实时数据来源：网销平台、 APP、官网、微信公号、 EPOS、微信客服、呼叫中心、掌 e 通、互联网公开数据。 实时采集方式：前端或后端埋点、爬虫方案批量采集数据内容：保单数据、理赔数据、保全数据、核保数据、财务分析数据。批量数据来源：关系型数据库、 Excel、 xml、 JSON、日志文件、音频、视频文件等。批量采集方式：全量或增量同步。

2、数据导入

实时数据采用 Flume+ Kafka+spark streaming, 批量数据采用 sqoop

3、数据清洗及标准化

对多源多格式的数据进行标准化处理，摒弃一些无效的数据。根据客户号、身份证号、港澳通行证、台湾同乡证、军官证、护照、驾照、车牌号、 IP 地址、电话、 mac 地址、邮箱、银行卡等数据对客户身份进行归一化处理。

4、建立标签及客户画像

建立多维标签，进行数据建模，进行客户 360 度画像。

5、 机器建模

将客户的各种标签属性和行为特征交给精准营销机器学习模型进行建模和训练，针对不同的场景建立客户购买意愿评分、产品推荐、客户忠诚度等模型。由于营销渠道、营销场景及营销目标等方面存在差异，可以根据情况建立不同的模型。

6、 模型应用及验证

配合模型设计 3 个月的营销活动，根据机器学习模型的预测目标，对客户进行探索，评估是否满足模型要求，并对模型的有效性进行验证。

7、 模型迭代优化

营销活动结束后，分析模型的实际预测效果，并对模型进行优化。模型的优化需要

不断整合更新的数据，让机器学习模型能够使用更新的客户数据做进一步的模型训练。

模型的优化过程是一个持续不断、周而复始的过程。

**四十六、请用图形方式描述云服务架构的三个层次分别是什么？保险企业一般重点建设的是哪**个层次的云架构？

**基础架构云 IaaS**、 应用平台云 PaaS 和业务服务云 SaaS 是云服务提供的三种层次，最基础的是 IaaS，中间的为 PaaS，最后直观呈现出来的是 SaaS。

1) IaaS 基础设施即服务

（Infrastructure-as-a-Service）

• 拥有了 IaaS，就可以将引荐外包到别的地方去。 IaaS 会提供场外服务器，存储和网络硬件，也可以选择租用。节省了维护成本和办公场地，公司可以在任何时候利用这些硬件来运行其应用。

• 目前比较知名的 IaaS 公司有亚马逊、 Bluelock、 CSC、 GoGrid、 IBM 等。

2) PaaS 平台即服务（Platform-as-a-Service）

• 某些时候也被叫作中间件。所有的开发都可以在这一层进行，节省时间与资源。 PaaS可以提供各种开发和分发应用的解决方案，比如虚拟服务器和操作系统等，可以节

省硬件上的费用。

• PaaS 公司与 IaaS 公司有许多重叠，除了上面列出的那些之外，还有 Google、Microsoft、Azure、 Force.com、 ,Heroku、 Engine Yard 等。

3) SaaS 软件即服务（Software-as-a-Service）

• SaaS 是我们目前普通用户接触最多的层面，在网络上任意一个远程服务器上的应用都是属于 SaaS。

• 比如现在阿里的钉钉、 JIBUU 以及苹果的 iCloud 都属于这一类。比较知名的 SaaS 公司有 Salesforce、 workday、 Slack 等。

目前对于保险企业来说，主要建设的是 PaaS 云平台。