**1.2 什么是分布式计算系统？它的实质是什么？**

分布式计算系统是由多个相互连接的计算机组成的一个整体，这些计算机在一组系统软件（分布式操作系统或中间件）环境下，合作执行一个共同的或不同的任务，最少依赖于集中的控制过程、数据和硬件。

实质：分布计算系统=分布式硬件+分布式控制+分布式数据。

**1.10多处理机与多计算机的区别是什么？同构多计算机和异构多计算机各有什么特点？**

区别：多计算机是将多个计算机联合起来处理问题，多处理机是在一个系统内集成多个处理器.

广义上说，使用多台计算机协同工作来完成所要求的任务的计算机系统都是多处理机系统。即多计算机系统。

狭义上说：多处理机系统的作用是利用系统内的多个CPU来并行执行用户的几个程序，以提高系统的吞吐量或用来进行冗余操作以提高系统的可靠性。

同构计算机的特点：

1.每个节点是一台计算机，包含CPU和存储器。

2.节点间的通信量较少。

3.同构计算机系统的互连有两种结构：基于总线的多计算机系统和基于交换的多计算机系统。

异构计算机的特点：

1.节点可能是多处理机系统、集群或并行高性能计算机。

2.节点间通过互联网络如Internet连接起来的。

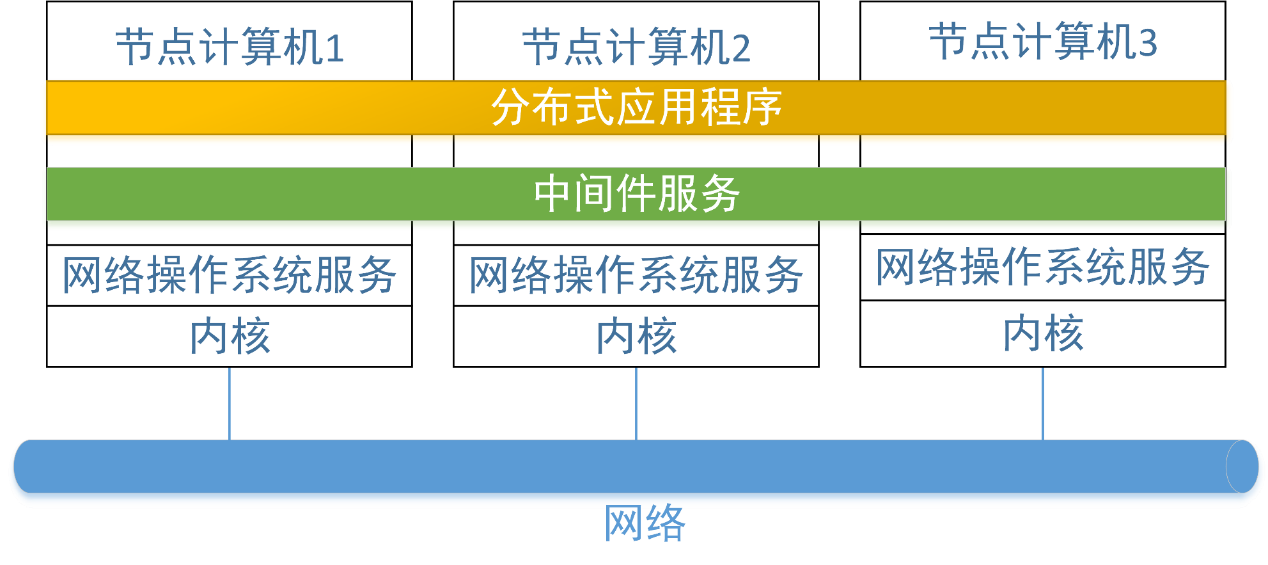
3.有两种实现方法：采用分布式操作系统和中间件软件层。

**1.16什么是中间件，它的功能是什么？它在分布式系统中的地位是什么？**

中间件是一种独立的系统软件或服务程序，分布式应用软件借助这种软件在不同的技术之间共享资源。中间件位于客户机/服务器的操作系统之上，管理计算机资源和网络通讯，是连接两个独立应用程序或独立系统的软件

功能：命名服务 作业调度 高级通信服务 资源管理 数据持久化 分布式事务 分布式文档系统 安全服务

地位：中间件的一个重要目标是对应用程序隐藏底层平台的异构型，因此中间件系统都提供一组完整度不同的服务集。这些服务是通过中间件系统提供的接口来调用的。一般禁止跳过中间件层直接调用底层操作系统的服务。



**1.18分布式系统有哪些计算模式？（必考）**

1.面向对象模式OOM（ Object Oriented Model）是基于客户/服务器模型（如CORBA common object request broker architecture公用对象请求代理结构，DCOM distribute component object model)

2.面向服务模式SOM （Service Oriented Model）

Web Service是这种面向服务模式的一个实例 ，SOA是一个较完整的软件结构体系。

3.公用计算模式UBM（Utility Based Model）

4.志愿参与模式VJM(Voluntary Join Model)

（详见书p21-p22页）

**2.5有哪些名字服务形式？名字服务器的组成与功能是什么？**

名字服务形式：

（1）名字服务：名字服务是根据实体的名字查找它的属性（地址）。**白页服务**

（2）目录服务：目录服务既可以根据实体的名字查找实体的属性，当不知道实体名时也可以根据实体的一个或多个属性及其值查找并得到一个匹配这些属性的实体列表。**黄页服务**

（3）合约服务：是一种增强的目录服务，通过技术规范来定位一个命名实体。**绿页服务**

名字服务器组成：

（1）名字服务器操作：管理、查询操作和行政管理。增加、删除和修改上下文的目录项。访问优先权。

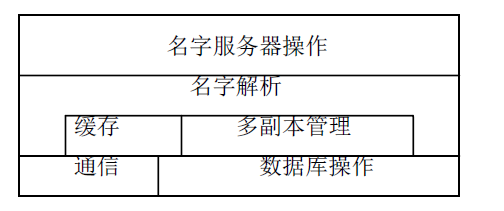
（2）名字解析 ：根据名字解析请求，得到被解析对象地址。

（3）缓存 ：缓存名字查询和解析的结果。

（4）多副本管理 ：副本修改和副本一致性维护。

（5）通信 ：客户端的名字代理通信和名字服务器之间

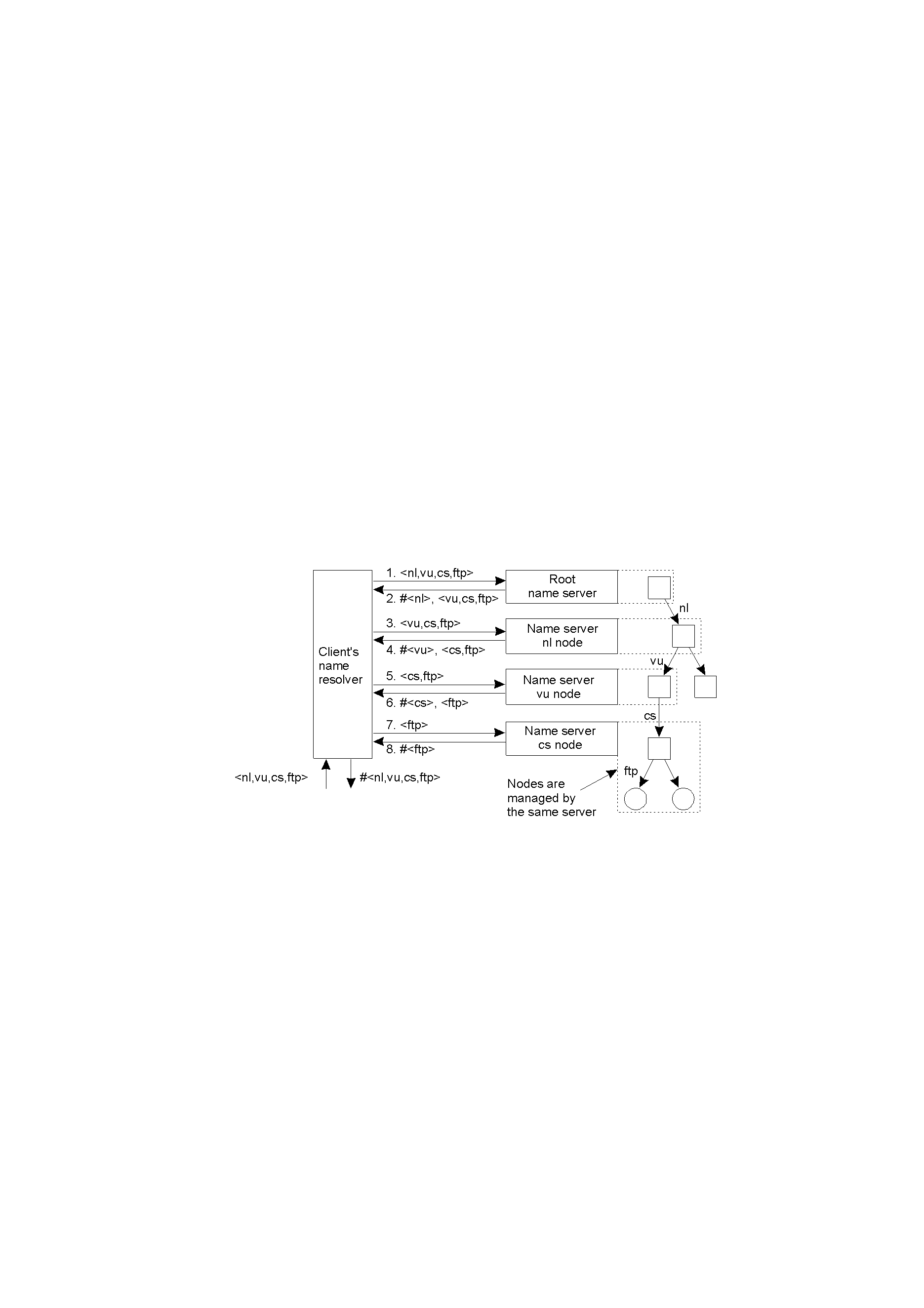
（6）数据库 ：存放名字解析上下文或其子域。



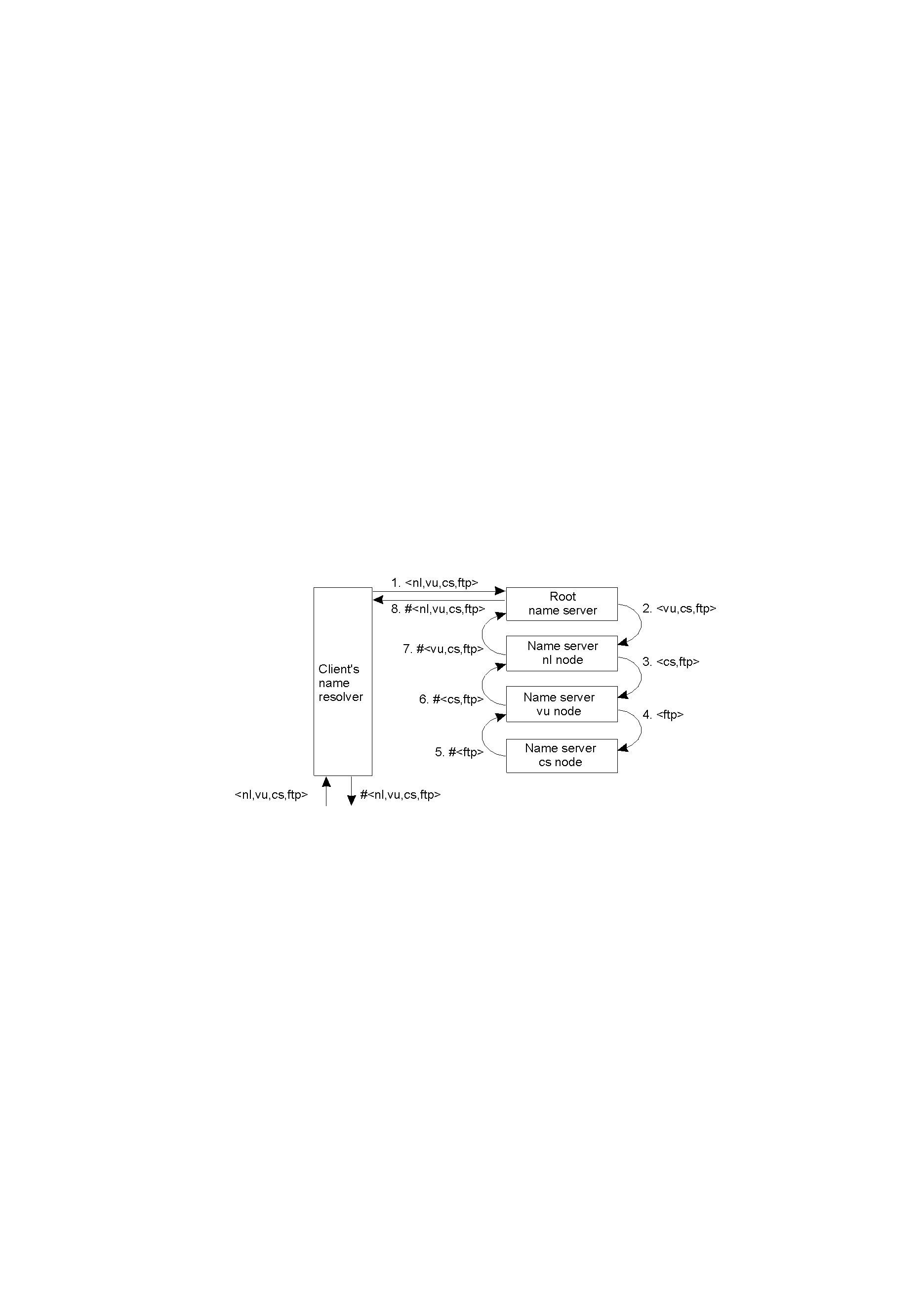
名字服务器功能：管理名字解析上下文、实现名字查询与解析和其它名字服务器通信协调。

**2.7什么是迭代名字解析，什么是递归名字解析，它们各有什么优缺点？**

迭代名字解析：建议考试画图解释：



递归名字解析：也画图解释



各自优缺点：

递归名字解析缺点：要求每台名字服务器具有较高的性能。

递归名字解析优点：1.递归名字解析过程中，各个名字服务器解析的缓存结果使用更为高效。2.如果主机与服务器距离很远，那么采用递归名字解析将更为高效。

迭代的优缺点与上面相反。

**2.14什么是目录服务？目录项和属性及属性值的关系是什么？**

目录服务：目录服务既可以根据实体的名字查找实体的属性，当不知道实体名时也可以根据实体的一个或多个属性及其值查找并得到一个匹配这些属性的实体列表。

关系：目录项是一个命名对象的信息集合。每个命名对象包括若干个属性，每个属性有一个属性类型和相应的一个或多个属性值。

目录信息库DIB(Directory Information Base)

目录系统代理DSA(Directory System Agents)

目录用户代理DUA(Directory User Agents)

目录管理域DMD(Directory Management Domain)

目录访问协议DAP(Directory Access Protocol)

目录系统协议DSP(Directory System Protocol)

目录信息镜像协议DISP(Directory Information Shadow Protocol)

目录操作绑定管理协议DOP(Directory Operaional Binding Management Protocol)

**2.17X.500目录服务中定义了哪些目录服务协议？查询链与转交的含义是什么？**

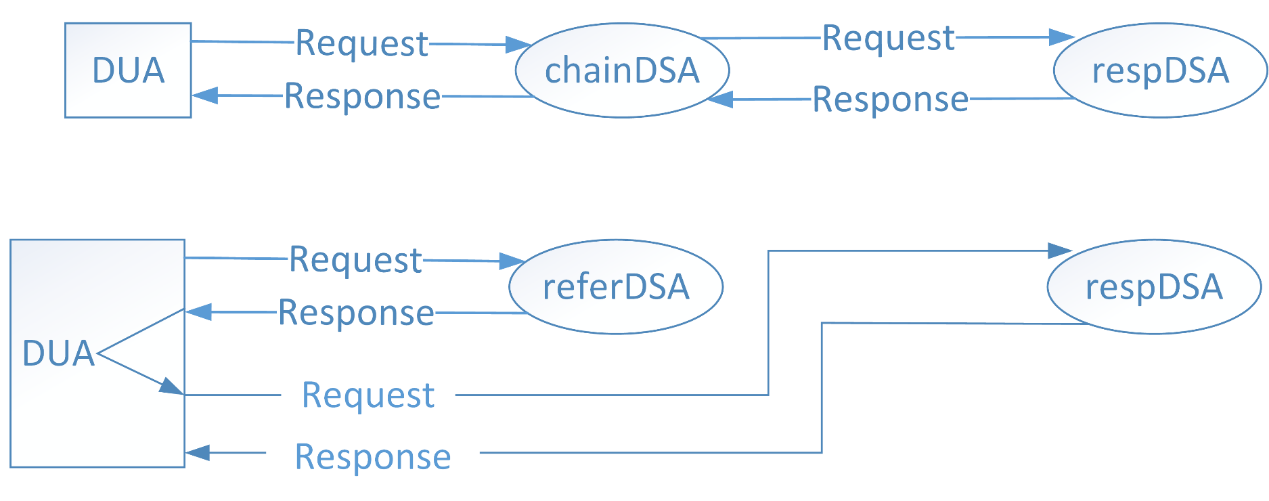
目录服务协议：

目录访问协议DAP，DUA用来与DSA通信。

目录系统协议DSP，是两个DSA之间的操作协议，在DSA之间传递查询请求和响应。

目录信息镜像协议DISP，是DSA用来将信息从镜像提供者传送给镜像使用者。

目录操作绑定管理协议DOP，DSA用来层次操作绑定管理和镜像管理。



查询链：单独一个DSA不太可能满足查询要求，它可能将一个查询转发到另一个DSA。

转交：当DSA发现整个查询操作必须由一个更适合的DSA处理时，它会返回一个转交(Referral)给上一个DSA或目录用户代理，这个转交指出更合适DSA的名字和地址。

**2.18轻量目录访问协议LDAP** **(Lightweight Directory Access Protocol)和目录访问协议DAP的关系和区别是什么？**

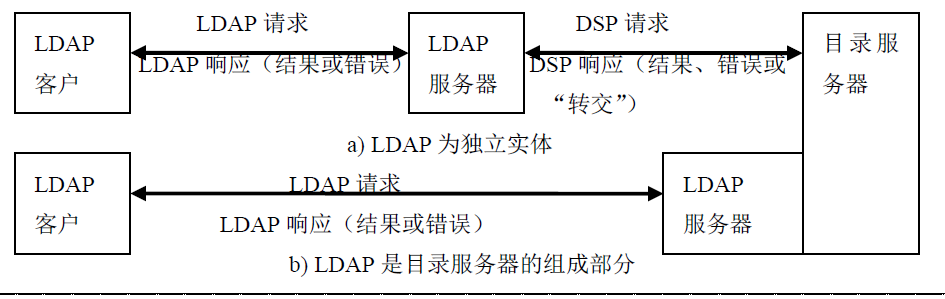
1.LDAP的最初目标是向用户提供目录服务时避免DAP的大量开销。

2.LDAP的操作集对DAP做了简化，删除了read和list操作，用search代替。

3.DAP是目录用户代理（DUA）与目录系统代理（DSA）之间的请求/响应协议。

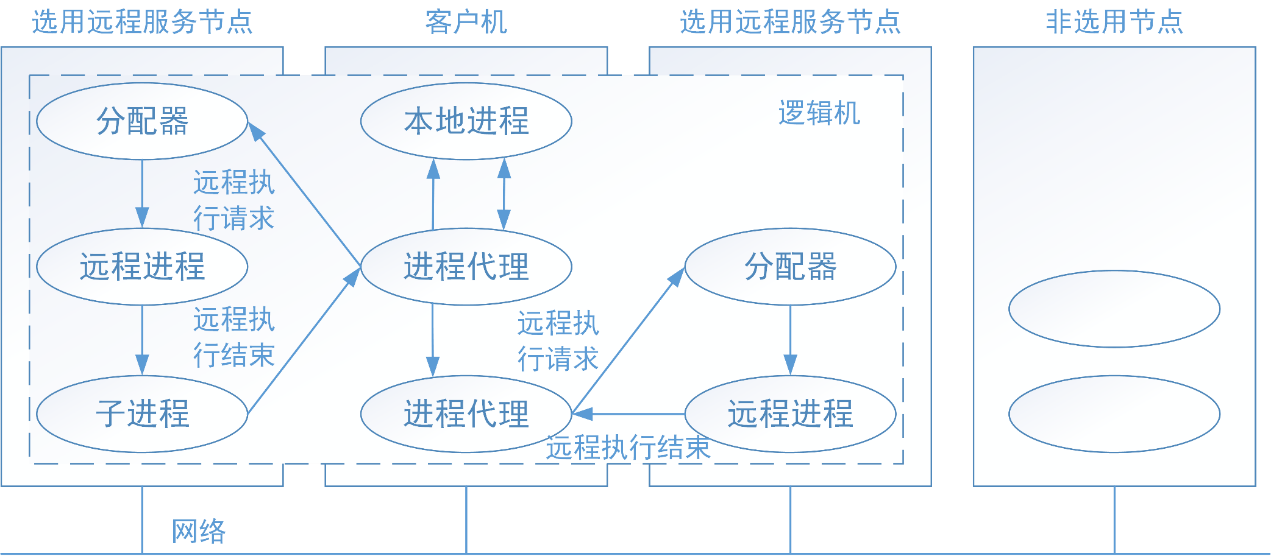
LDAP是用户用来访问目录服务的一个协议。

4.建议再回答下LDAP的模型：



**3.7什么是远程执行逻辑机模型？对逻辑机模型的要求是什么？**

概念：建议画图。



要求：

（1）远程进程必须能访问驻留在源计算机上的文件系统。

（2）远程进程能接收逻辑机内任何进程发来的信号，也能将信号提供给逻辑机内任何进程。

（3）进程组保持在逻辑机内。

（4）基于树型的进程父子关系在逻辑机内必须得以保持。

**3.13何为异步进程迁移算法？何为同步进程迁移算法？它们的优缺点是什么？**

异步进程迁移算法：这类算法允许非迁移进程在迁移过程中继续运算，只有迁移进程被中断进行相关的操作。

优点：可以得到较好的执行效率。

缺点：和原有环境的兼容性不好，不能方便的移植。

同步迁移算法：这类算法在迁移过程中所有进程（包括非迁移的协同进程）都被挂起，进程之间需要同步来清空通信信道中的中途消息，所有进程均要阻塞等待迁移事件完成后，才能从中断处继续运行。

优点：算法简单，具有较好的可移植性和易于实现。

缺点：需要中央控制管理进程参与，所有进程都被迫中断，等待迁移过程的结束。

**3.15比较进程远程执行与进程迁移两种机制。**

进程远程执行的要求：

（1）寻找管理机制。

（2）进程远程执行是透明的，应与位置无关。

（3）主人优先原则

进程的迁移可以支持：

（1）动态系统管理与维护

（2）动态负载平衡（load balancing），系统中重负载处理机转移一部分负载到轻负载的处理机上运行，使得整个集群系统中的所有处理机的负载趋向均衡，从而提高系统的整体运行效率。

（3）系统容错

（4）主人优先使用原则

注：此题我未找到较好的答案。

**4.1在水平时间轴上表示阻塞发送/接收和非阻塞发送/接收进程与操作系统内核之间操作的时间关系。**

没有具体答案，先方便理解一下阻塞和非阻塞：

阻塞和非阻塞关注的是程序在等待调用结果（消息，返回值）时的状态.

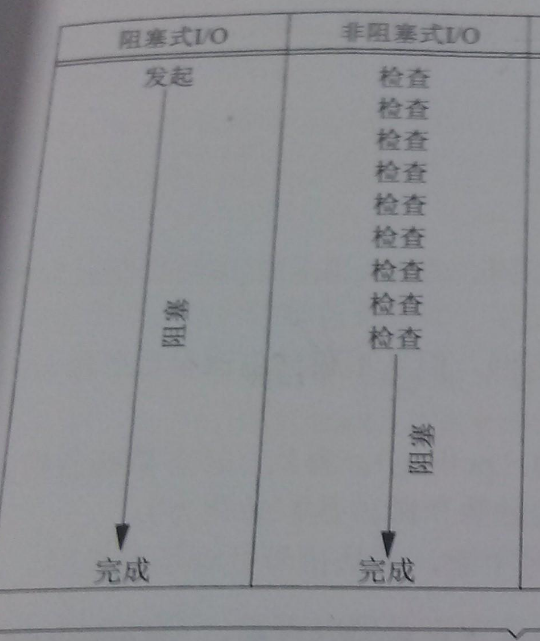
阻塞调用是指调用结果返回之前，当前线程会被挂起。调用线程只有在得到结果之后才会返回。

非阻塞调用指在不能立刻得到结果之前，该调用不会阻塞当前线程。

例子：

你打电话问书店老板有没有《分布式系统》这本书，你如果是阻塞式调用，你会一直把自己“挂起”，直到得到这本书有没有的结果，如果是非阻塞式调用，你不管老板有没有告诉你，你自己先一边去玩了， 当然你也要偶尔过几分钟check一下老板有没有返回结果。

这个图不知道对不对：



**4.2试叙述如何实现阻塞发送/接收和非阻塞发送/接收，对操作系统有什么要求？**

通过发送原语send和接收原语receive实现要求操作系统能实现4种不同的可靠性语义。

至少一次：保证正确完成消息传送至少一次

至多一次：保证正确完成消息传送至多一次。在没有节点崩溃和网络断开情况下，它只正确地执行一次消息传送。

事务语义：它保证消息的原子性。不管节点崩溃或网络端口与否，它或者完成一次消息传送，或者什么也不做。

精确一次：无论在什么情况下，保证正确完成一次消息传送，不管是否有节点崩溃或网络断开，它接近某种程度的容错机制。

**4.4对以下每个应用程序，你认为“至多一次”和“至少一次”语义哪个最好？**

（1）在文件服务器上读写文件：至少一次。

（2）银行服务：至多一次

（3）编译一个程序：至少一次

**4.9什么是因果定律？它和FIFO全定序相比，哪个更严格？**

因果定律：不管含有因果关系的消息是由同一个发送进程多播，还是不同发送进程多播，所有接收进程要保证先接收“因”消息，后接收“果”消息。

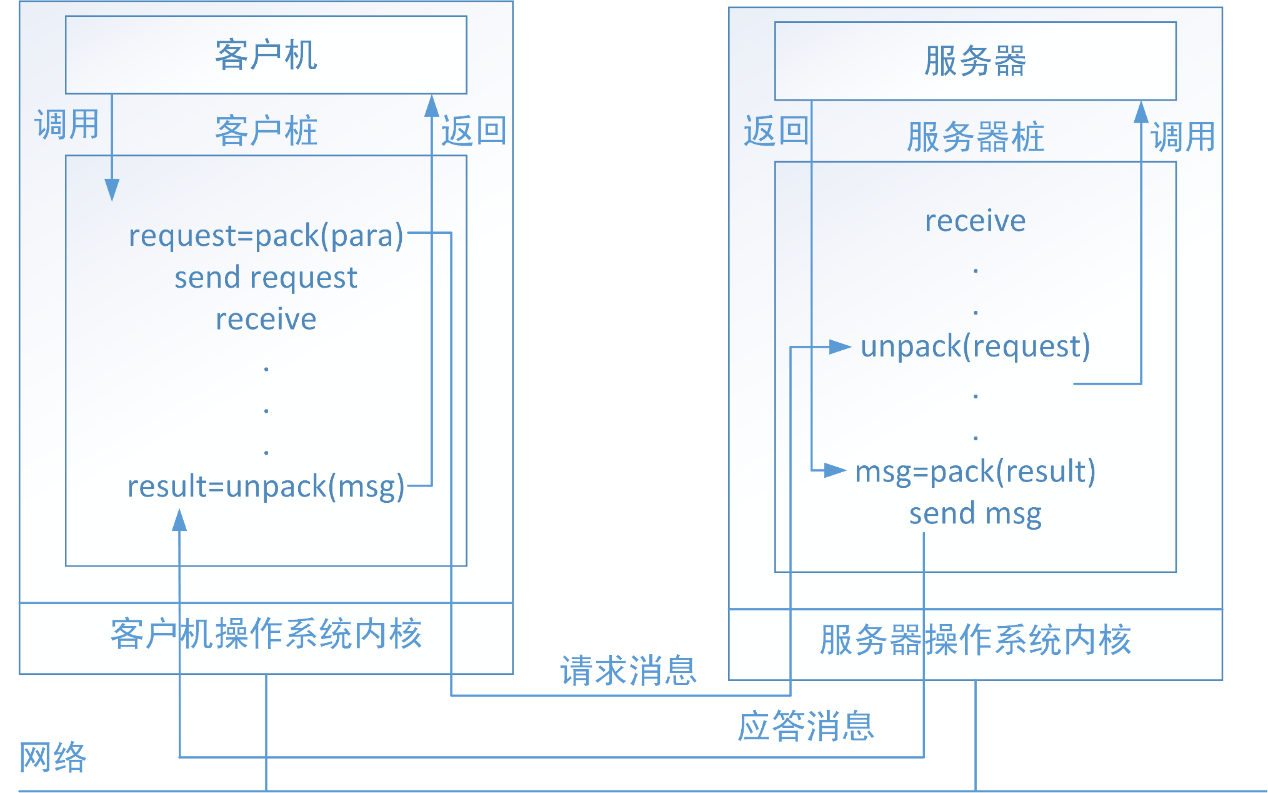
FIFO：对同一个发送进程发出的多播消息，要求所有接收进程按发送的顺序接收，而对不同发送进程的多播消息可按不同顺序接收。

相比之下，因果排序更严格。

**4.12RPC（Remote procedure call 远程过程调用）被认为是分布式最初的中间件，它能实现分布式系统的透明性吗？**

p92 在RPC过程中，客户可以简单的忽略不关心的内容，客户并不直接执行send和receive原语，也不关心消息的传递，所有这些都隐藏在桩中，从而实现RPC的透明性。

建议；本题虽然已完，但最好将RPC的基本操作看一遍。



**5.2假设两台机器的时钟每秒滴答1000次和990次，如果UTC每秒更新一次，两台机器时钟的最大偏移量是多少？**

1000-990=10次/秒 每秒最大偏移量0.010ms

**5.6在集中式互斥算法中，若考虑进程的优先权，算法应如何设计？**

对进程的服务按优先权进行

适应优先权系统（进程优先权，P系统）和实时系统（进程执行残留时间，RT系统）的两种互斥算法

完全可靠网络算法

请求队列P、Q

Q队列放置其他节点送来的请求（接收令牌）

P队列放置其他节点来不及处理的随令牌转来的请求

算法过程

i 希望进入，发送Request（i，P（i）），并将（i，P（i））存入接收接收进程的Q队列

j 退出 根据P、Q队列情况判断（标注最高优先权进程，合并队列）

j 将令牌和新的P队列发送到所标注的最高优先权进程

不可靠网络算法

节点（进程）失效

消息丢失或不按顺序发送

不可靠网络下，分布式互斥算法

最多一个竞争进程进入临界区，且机会均等（定时）

按进程优先关系而不是按它出现的先后被批准

应有节点失效检测恢复机制

进程进入和退出竞争进程组的开销小，不增加消息数

算法基本与可靠网络互斥算法一致

队列P不空而Q空，进程将令牌和P队列发送指定进程，并启动超时机制

在超时范围内，如果没有收到指定进程的应答，进程从队列删除该进程，同时将令牌发给下一个进程。

**5.7 Richart\_Agrawala算法如何改进了Lamport算法，它的优点是什么？（必考）**

Lamport算法的开销是3（N-1）个消息，Richart算法只要2（N-1）个消息，N是竞争资源的进程数。

Lamport算法前2步：

1.Pi 资源请求消息Request（Ti ：P i）发送

2.Pj收到Request（T i：P i），按T顺序置于其消息队列，如果没有资源请求或请求时间晚于收到消息的时间戳，回应Reply（T j：P j）

Richart算法前2步：

1.Pi 资源请求消息Request（Ti ：Pi）发送

2.Pj收到Request（Ti：Pi），按T顺序置于其消息队列

如果没有资源请求或请求时间晚于收到消息的时间戳，回应Reply（Tj：Pj）；否则推迟返回应答消息

进程从临界区退出，向需要请请求资源的进程补发应答消息

标红处即为改进

优点：

1.它具有对称性

2.具有完全的分布式控制

3.对通信链路相对速度的不敏感性

**5.8比较集中式算法、Ricart\_Agrawala算法和令牌算法的开销和问题**

集中式算法的问题：容易出现单点故障

Ricart算法的问题：由于不应答被认为是资源被占用，所以如果有某个节点故障，会导致该算法的异常终止。同时各进程对资源的使用情况缺乏了解。

令牌算法的问题：检测令牌丢失困难

开销：集中式算法开销最大，Richar算法需要2（N-1）个消息，令牌算法最多需要N-1个消息。

**5.11共享K个相同资源的互斥算法和Ricart\_Agrawala算法的共同点和区别是什么？**

相同点：基于相同的概念，每个竞争进程都维持一个推迟应答数组RD[]，数组元素是表示相应进程是否推迟发出应答消息。

区别：

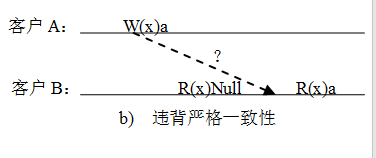
1.应答消息到达的环境。在Ricart算法中，正在等待进入临界区的进程要得到N-1个应答消息。在共享K个相同资源的互斥算法中，N-K个应答消息是在进程等待时到达，K-1个消息是进程已在临界区或等待进入临界区或离开临界区后到达。

2.在Ricart算法中，其他竞争进程推迟应答数组的每一项RD[i]是布尔型，因为应答只能是一个，或是推迟，或是不推迟。在共享K个相同资源的互斥算法中，可能有多个应答消息被推迟，这样RD[i]应声明为整数型。

**5.13在基于事件优先权算法中，如何保证低优先权的进程有机会进入临界区，而不挨饿。**

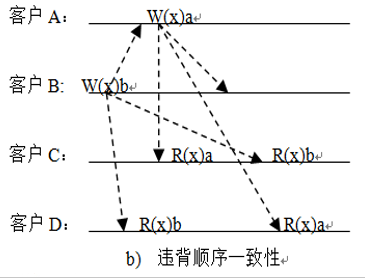
可参考Ricart算法

**8.2图8.1（b）为什么违背严格一致性？**



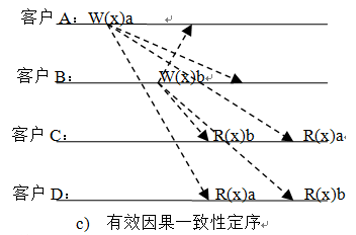
因为B读到的不是a，而是数据项x的初值null，客户A的写操作没有立即传播到B，未能及时完成对副本的修改。

**8.3图8.2（b）为什么违背顺序一致性？**



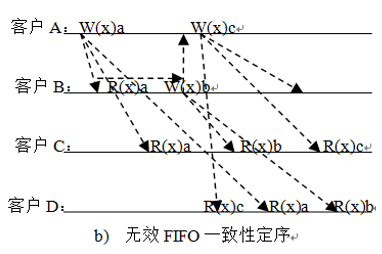
因为进程C看到数据项x是先写a后写b，而进程D看到数据项x是先写b后写a。

**8.4图8.3（c）为什么符合因果一致性定律？**



因为W（x）a和W（x）b是并发写，因此也符合因果一致性要求。

**8.5图8.4（b）为什么遵循因果一致性，但对FIFO一致性是无效的？**



客户C对数据项x读a，b，c;而客户D对读c，a，b虽然遵守因果一致性，但不符合FIFO一致性，

**8.7比较“传播更新通知”、“传播更新数据”和“传播更新操作”和它们的应用场合**

传播更新通知：只传播一个简短的数据通知，不包含其他信息。写操作对读操作的比率很高时，传播效果好。

传播更新数据：在副本间传送被修改过的数据。读操作对写操作的比率很高时，传播效果好。

传播更新操作：不传播被修改过的数据，而是告诉各副本应该执行的操作。要求每个副本有一个进程来执行更新操作，主动的保持各副本关联数据的一致性。

应用场合：

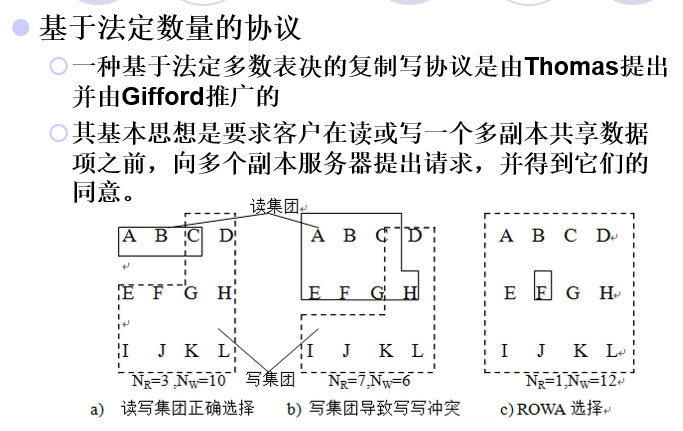
传播更新通知：无效化协议

传播更新数据：读操作对写操作的比率很高时

传播更新操作：主动复制

**8.9一个文件被复制在10个服务器上，列出表决算法的“读集团”和“写集团”。**

这是12个服务器是的例子：



此算法要求N（R）+N（W）>N

N（W）>N/2

此题的N=10;

那么写集团可以为6 7 8 9 10

相应的读集团为5 6 7 8 9 10 （NW=6）

4 5 6 7 8 9 10(NW=7)

以此类推。

**8.14在基于页面的DSM中如何维护副本拥有者和副本集这类控制信息？**

拥有4种算法：

集中管理算法

固定分布管理算法

广播分布管理算法

动态分布管理算法

在书p223-225，此处不详述。

**9.2分布式文件系统的共享语义指的是什么，有哪几种共享语义？**

文件共享语义是为了理解文件的行为。不同共享语义将导致不同编程实现方法。

种类：

（1）UNIX语义（顺序一致性语义）

（2）会话语义

（3）不修改共享文件语义

（4）事务语义

见书p242

**9.4NFS如何访问远程文件，它的虚拟文件系统的作用是什么？**

如何访问：远程共享文件或它的子目录是通过虚inode（即vnode）形式挂接在本地文件目录中。NFS客户端与远程文件系统服务器之间通信采用安全的远程过程调用（RPC）方式，实现身份验证和消息验证，采用了RPC重传技术，对文件共享进行访问控制。

作用：1.允许和操作系统使用不同的文件系统实现的接口。

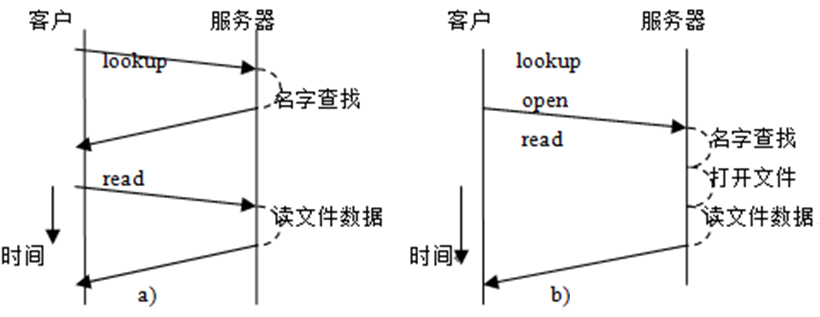
2.是物理文件系统与服务之间的一个接口层，对每个文件系统的所有细节进行抽象，使得不同的文件系统在系统中运行的其他进程看来，都是相同的。

**9.8NFS用的是远程访问模式还是上载下载模式？**

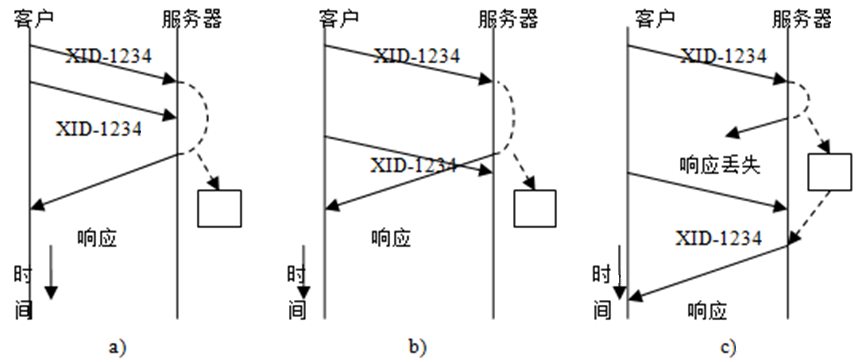
远程文件访问模型

**9.9NFS的RPC实现了何种可靠性语义？**

通过复合调用



以及RPC重传



可靠性语义有：

至少一次：保证正确完成消息传送至少一次

至多一次：保证正确完成消息传送至多一次。在没有节点崩溃和网络断开情况下，它只正确地执行一次消息传送。

事务语义：它保证消息的原子性。不管节点崩溃或网络端口与否，它或者完成一次消息传送，或者什么也不做。

精确一次：无论在什么情况下，保证正确完成一次消息传送，不管是否有节点崩溃或网络断开，它接近某种程度的容错机制。

我们可以看出：符合“至少一次”的可靠性语义。

**11.5CORBA的主要特点是什么？功能是什么？简述CORBA系统中ORB的作用。**

特点：

（1）CORBA定义了一种面向对象的软件构件构造方法，使不同的应用可以共享由此构造出来的软件构件；

（2）每个对象都将其内部操作细节封装起来，同时又向外界提供了精确定义的接口，从而降低了应用系统的复杂性，也降低了软件开发费用；

（3）CORBA的平台无关性实现了对象的跨平台引用，开发人员可以在更大的范围内选择最实用的对象加入到自己的应用系统之中；

（4）CORBA的语言无关性使开发人员可以在更大的范围内相互利用别人的编程技能和成果，是实现软件复用的实用化工具

功能：

（1）存取来自现行桌面应用程序的分布信息和资源；

（2）使现有业务数据和系统成为可供利用的网络资源；

（3）为某一特定业务用的定制的功能和能力来增强现行桌面工具和应用程序；

（4）改变和发展基于网络的系统以反映新的拓扑结构或新资源；

ORB的作用：

ORB初始化 ORB接口操作 获取初始对象引用 与线程有关的操作 策略设置与管理 对象引用操作

**11.6什么是接口定义语言？它在分布式计算环境中起到什么作用？**

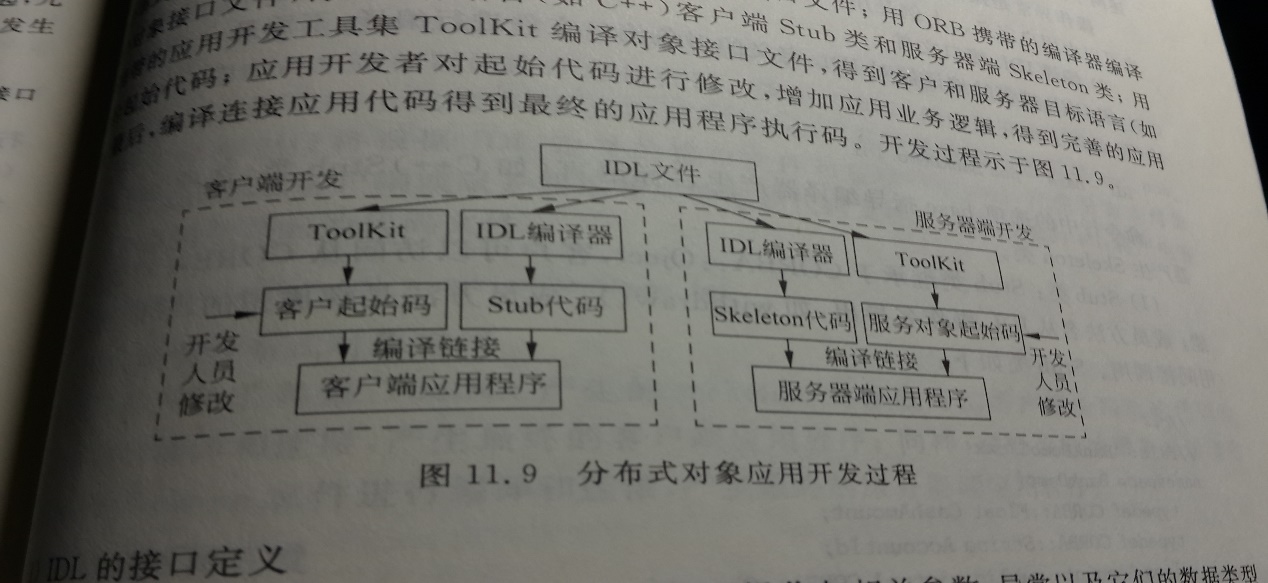
概念：是一个描述软件组件接口的语言规范。 IDL用中立语言的方式进行描述，能使软件组件（不同语言编写的）间相互通信。 IDL通常用于RPC（Remote Procedure Call，远程过程调用）软件。 IDL提供了一个“桥”来连接不同的系统。

作用：

（1）用IDL定义对象接口文件

（2）用ORB携带的编译器对IDL定义的对象接口文件进行编译。

然后将此图画上：



**11.7在CORBA的体系结构中，根据接口的可移植性和定义方式，接口分为哪几类，各起什么作用？**

1.ORB核心

实现对所有ORB都相同的接口

2. 对象适配器接口

连接对象实现和ORB。

3. 接口定义语言IDL和静态接口

客户用ORB携带的IDL编译器，编译对象接口IDL文件，生成特定编程语言（如C++）的Stub和skeleton代码。

4. 动态接口

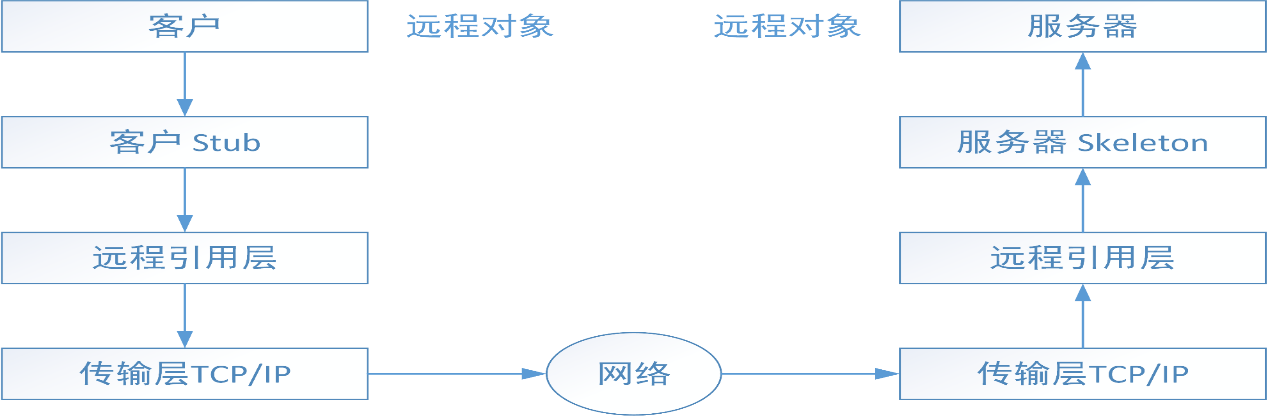
在预先不知道服务对象接口的情况下，客户通过查询或采用其他手段获得服务对象的接口描述信息，然后使用动态调用接口来构造客户请求，并发送给对象实现。

在对象实现方，可用动态骨架接口分发用户请求的机制，以便动态处理客户方的请求。

**11.17简述JAVA RMI体系结构**

远程方法调用RMI（Remote Method Invocation）是Java特有的分布式计算技术。它实质上是通过Java编程语言扩展了常规的过程调用，在网上不仅可以传送对象的数据，而且可以传送对象的代码。

回答下面这张图即可：



**12.6什么是web契约，它要说明什么问题？**

概念：契约是供求双方间进行交换的一种约定，在面向服务的分布式计算系统中，契约是系统之间交换数据时应遵守的约定（技术契约）。

说明的问题：服务功能描述（What） 服务访问描述（How）

服务位置描述（Where）

**12.7什么是服务的抽象描述？什么是服务的具体描述？它们描述的内容是什么?**

抽象描述：即服务功能描述，表达了契约公开的接口

具体内容：

端口类型（接口）定义（portType）

操作定义（Operation Definition）

消息定义（Message Definition）

类型定义（Type Definition）

策略定义（Policy Definition）

具体描述：即服务访问描述和服务位置描述，为抽象描述补充相关的实现细节，即如何访问服务，同时关注从何处得到所需要的服务。

具体内容：

端口类型（接口）绑定（Port Binding）

操作绑定（Operation Binding）

消息绑定（Message Binding）

服务定义（Service Definition）

端口定义（Port Definition）

地址定义（Address Definition）

策略定义

**12.8什么是WSDL？它与XML有什么关系？WSDL用来描述什么实体？**

WSDL:即web服务描述语言。

关系：

1.WSDL是一个基于XML的语言，它描述了和特定Web服务之间的交互机制，并且使用该语言可以约束服务提供者以及使用服务的所有请求者

2.WSDL是一个基于XML的规范模式，提供了一个标准的服务表示语言，可用于描述Web服务所暴露的公共接口细节

WSDL用于描述网络服务。

**12.9什么是SOAP？它与HTTP和WSDL有什么关系？**

SOAP:简易对象访问协议。

关系：SOAP是个通信协议，SOAP在HTTP协议的基础上，把编写成XML的REQUEST参数，放在HTTP BODY上提交给WEB SERVICE服务器。WEB SERVICE服务器处理完成后，把结果也写成XML作为RESPONSE送回用户端，为了使用户端和WEB SERVICE可以相互对应，可以使用WSDL作为这种通信方式的描述文件，利用WSDL工具可以自动生成WS和用户端的框架文件。

**12.10**什么是UDDI?它与HTTP、WSDL和SOAP的关系是什么？

UDDI:通用描述、发现和集成

关系：

WSDL用来描述服务；

UDDI用来注册和查找服务；

SOAP作为传输层，用来在消费者和服务提供者之间传送消息。SOAP是Web服务的默认机制，其他的技术为可以服务实现其他类型的绑定。

用户可以在UDDI注册表（registry）查找服务，取得服务的WSDL描述，然后通过SOAP来调用服务，同时UDDI采用的也是HTTP协议。

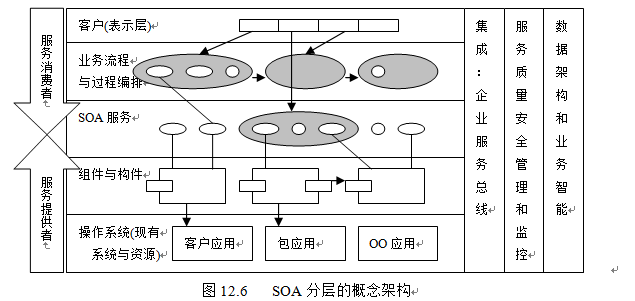
或答：

Web服务提供方通过WSDL描述所提供的服务，并将这一描述告知Web服务注册服务器。注册服务器依据WSDL的描述，依照UDDI的协定更新服务目录并在Internet上发布。用户在使用Web服务前先向注册服务器发出请求，获得Web 服务提供者的地址和服务接口信息，之后使用SOAP协议与Web服务提供者建立连接，进行通信。

**12.13什么是SOA概念框架？SOA试图解决什么问题？**

SOA的概念架构采用分层模式，这个架构自底向上是操作系统层、服务组件层、服务层、业务流程编排层和访问表现层。

把图画上：



试图解决的问题：

（1）企业业务模式的变化：传统的业务部门的消失，如企业运输部门

（2）过去的IT系统建设以部门为基础整合，是部门内的垂直整合；现在需要在企业各部门间进行水平整合

（3）企业IT系统抽象程度低

（4）企业IT系统改变或者升级时，原有的硬件和软件资源希望在新系统中尽可能重用

**12.17试述BPEL、SCA和SDO之间的关系**

1. 服务编排和流程整合

服务的串联，即业务流程BPEL、服务的构建和使用SCA

数据的描述SDO，这三部分构成了SOA架构铁三角

2. SOA中，需要有数据模型与其他的数据模式交互（如数据库，文件），与SCA相应的数据模型为SDO

3. SCA组件的实现类型包括BPEL