关于分布式系统复习题与参考答案

一、 填空题(每题 n 分,答错 个扣 分,全错全扣，共计 m 分)

1．下面特征分别属于计算机网络和分布式计算机系统，请加以区别：
分布式计算机是指系统内部对用户是完全透明的；系统中的计算机即合作又自治；系统可以利用多种物理和逻辑资源，可以动态地给它们分配任务。
计算机网络是指互连的计算机是分布在不同地理位置的多台独立的“自治计算机”。

2．点到点通信子网的拓扑结构主要有以下几种：星型、环型、树型、网状型，请根据其特征填写相应结构。
网状型 ：结点之间的连接是任意的，没有规律。环型：节点通过点到点通信线路连接成闭合环路。星型：节点通过点到点通信线路与中心结点相连；树型：结点按层次进行连接。

3．分布式计算系统可以分为两个子组，它们是集群计算系统和网格计算系统。

4．分布式事务处理具有 4 个特性，原子性：对外部来说，事务处理是不可见的；一致性：事务处理不会违反系统的不变性；独立性：并发的事务处理不会相互干扰；持久性：事务处理一旦提交，所发生的改变是永久性的。

5．网络协议有三要素组成，时序是对事件实现顺序的详细说明；语义是指需要发出何种控制信息，以及要完成的动作与作出的响应；语法是指用户数据与控制信息的结构与格式

6．根据组件和连接器的不同，分布式系统体系结构最重要的有 4 种，它们是：分层体系结构、基于对象的体系结构、以数据为中心的体系结构、基于事件的体系结构

7．在客户-服务器的体系结构中，应用分层通常分为 3 层，用户接口层、处理层和数据层。

8．有两种类型的分布式操作系统，多处理器操作系统和多计算机操作系统。

9．软件自适应的基本技术有 3 种，一是要点分离、二是计算映像、三是基于组件的设计。

10．DCE 本身是由多个服务构成的，常用的有分布式文件系统、目录服务、安全服务以及分布式时间服务等。

11．TCP/IP 体系结构的传输层上定义的两个传输协议为传输控制协议(TCP)和用户数据报协议(UDP)。

12．Windows NT 的结构借用了层次模型和客户/服务器两种模型。

13．常用的进程调度算法有先来先服务、优先数法和轮转法

14．进程的三个基本状态是就绪、执行、等待(阻塞)。

15．进程是 程序 在一个数据集合上的 运行过程 ，是系统进行资源分配和调度的一个 独立单位

16．进程四个特征是 动态性， 并发性， 独立性 ，异步性。

17．操作系统通常可以分为以下几种类型：批处理系统、分时系统、 实时系统 、 网络操作系统 和分布式操作系统。

18．解决死锁的基本方法包括预防死锁， 避免死锁，死锁检测，死锁恢复。

19．在引进线程的操作系统中，调度和分派的基本单位是线程，拥有资源的单位是进程。

20．在面向流的通信中，为连续提供支持数据流的模式有异步传输模式、同步传输模式和等时传输模式三种。

21. 在流同步机制，通常有在数据单元层次上进行显式同步和通过高级接口支持的同步两种。

22．在分布式系统中，挂载外部名称空间至少需要的信息是：访问协议的名称、服务器的名称和外部名称空间中挂载点的名称。

23．在名称空间的实现中，为了有效实现名称空间，通常把它划分为逻辑上的三层，其三层指的是全局层、行政层和管理层。

24．在名称解析的实现中，通常采用两种方法，一是迭代名称解析；二是递归名称解析。
25．在逻辑时钟算法中，Lamport 定义了一个称作“先发生”的关系，表达式 a→b 表示 a 在 b 之前发生。先发生关系是一个传递关系。
26．分布式系统中，通常有 4 种互斥算法，一是集中式、二是非集中式、三是分布式、四是令牌环。
27．分布式系统中的选举算法有两种，一是欺负选举算法；二是环选举算法。
28．在以数据为中心的一致性模型中，顺序一致性是指“任何执行结果都是相同的，所有进程对数据存储的读/写操作是按某种序列顺序执行的，并且每个进程的操作按照程序所制定的顺序出现在这个序列中”。
29．在因果一致性中，所有进程必须以相同的顺序看到具有潜在因果关系的写操作。不同机器可以以不同的顺序看到并发的写操作。
30．以客户为中心的一致性模型中，满足最终一致性的数据存储具有以下属性：没有更新操作时，所有副本逐渐成为相互完全相同的拷贝。
31. 以客户为中心的一致性模型中，一个写操作总是在同一进程执行的后续读操作之前完成，而不管这个后续的读操作发生在什么位置。
32．在一致性协议中，基于主备份的协议比较盛行，它包括远程写协议和本地写协议两种。
33．在一致性协议中，复制的写协议包括主动复制和基于多数表决的一致性协议两种。
34．在容错性中，故障通常被分为暂时性故障、间歇性故障和持久性故障三大类型。
35．如果系统是容错的，使用冗余掩盖故障的方法有信息冗余、时间冗余和物理冗余三种。
36．在可靠的客户-服务器通信中，失败时的 RPC 系统中发生客户不能定位服务器、请求消息丢失、服务器崩溃、应答消息丢失和客护端崩溃等 5 种形式。
37．在原子多播里，消息排序通常有 4 种不同的排序方法，它们分别是：不排序的多播、FIFO 顺序的多播、按因果关系排序多播和全序多播。
38．容错性的基本要求是从错误中恢复，本质上有两种形式的错误恢复，一是回退恢复；另一种是前向恢复。
39．在分布式安全性中，通常考虑计算机系统受到的安全威胁有窃听、中断、修改和伪造等四种。
40．安全策略准确地描述系统中的实体能够采取的行为以及禁止采取的行为。
41．安全机制包括加密、身份认证、授权和审计等四个部分。
42. 分布式加密系统通常有三种类型，一是对称加密系统（DES）；二是公钥加密系统（RSA）、三是散列函数（MDS）系统。
43．身份认证是一种会话密钥，常用的身份认证有基于共享密钥的身份认证、使用密钥分发中心的身份认证、使用公钥加密的身份认证三种类型。
44．消息的完整性是指保护消息免受修改；其机密性确保窃听者不能截获和读取消息。
45．在安全通道中，为了使消息完整性和机密性。通常采用数字签名和会话密钥的加密算法实现安全通道的数据交换。
46．在安全组通信中，要确保机密性，机密组通信使用公钥加密系统可以解决；而安全的复制服务器组中共享一个保密签名的方法。
47．在访问控制中，要建立主体对对象的访问权限，其普通方法是构造一个访问控制矩阵，而访问控制列表（ACL）和权能列表来实现。
48．防火墙通常分为两种不同的类型，一种是数据包过滤网关；另一种是应用层的网关。
49．在分布式系统安全管理中，主要分密钥管理、安全组管理以及授权管理三大内容。
50．在容错性中，人们定义了一些不同类型的故障，主要的有崩溃性故障、遗漏性故障、定时性故障、响应性故障以及随意性故障等五大类。

二、选择题(每题 n 分，共 m 个题，共计 x 分)

1．网络体系结构可以定义为：(C)

A、一种计算机网络的实现　　B、执行计算机数据处理的软件结构

C、建立和使用通信硬件和软件的一套规则和规范　　D、由 ISO 制定的一个标准

2．在 OSI 参考模型中，数据链路层的数据服务单元是：（ C ）

A、分组　　B、报文　　**C、帧**　　D、比特序列

3．下面属于分布式计算系统的是（B C）

A、资源管理　　**B、集群计算**　　**C、网格计算**　　D、运行应用程序

4．目前分布式信息系统按集成可分为（A B）

A、事务处理系统　　**B、企业应用集成**　　C、网络管理　　D、资源分配系统

5．现在已认可的分布式系统软件体系结构样式有（ABCD）

A、分层体系结构　　**B、基于对象的体系结构**

C、以数据为中心的体系结构　　**D、基于事件的体系结构**

6．两个旅行社甲和乙为旅客到某航空公司订飞机票，形成互斥的资源是(A)。

A．飞机票　　B．旅行社　　C．航空公司　　D．旅行社和航空公司

7．DNS 属于（ A ）层协议。

A．应用层　　B．传输层　　C．互联网层　　D．网络接口层

8．活动目录是一种（ AB ）结构的目录服务。

A．层次式　　**B．分布式**　　C．对等式　　D．主次式

9．对于域名：test.com，DNS 服务器查找顺序是（ B ）。

A．先查找 test 主机，再查找.com 域　　**B．先查找.com 域，再查找 test 主机**

C．随机查找　　D．以上答案皆是

10．SMTP 协议是关于（ A ）的协议。

A．邮件传输　　B．文件传输　　C．超文本传输　　D．网络新闻组传输

11．POP3 协议是关于（ C ）的协议。

A．超文本传输　　B．邮件传输　　**C．接收邮件**　　D．网络新闻组传输

12．远程客户端登录终端服务器必须提供一定的信息，下列有（AC）属于这种必要的信息。

A．用户名　　B．域　　**C．服务器 IP 地址**　　D．连接名称

13．在多播通信中，应用层多播树的质量通常以（ABC）不同的尺度来度量。

A．链接树　　**B. 相对延时补偿**　　**C. 树成本**　　D. 树结构

14．以多播流方式传递内容时只能采用（B）类型的发布点。

A．单播发布点　　**B．广播发布点**

C．单播发布点或广播发布　　D．既不是单播发布点也非广播发布点

15．DNS 名称空间是分层组织的一棵有根树，标识符是有（ C ）。

A．字母组成　　B. 数字组成　　**C. 字母和数字组成**　　D. 汉字组成

16．IDL 编译器的输出包括的文件是（ ABD ）。

A．文件头　　**B. 客户存根**　　C. 守护程序　　**D. 服务器存根**

17．下列属于流同步的是（C D）。

A．位同步　　B. 字同步　　**C. 离散数据流与连续数据流之间同步**

D．口型同步

18．实现线程包的基本方法有（AC）。

A．构造一个完全在用户模式下执行的线程库　　B. 由进程间通信实现

C．由内核来管理线程并进行调度　　D. 由用户程序来实现

19．下面是多线程服务器可行的设计方法是（ACD）

A．多线程文件服务器　　B. Web 服务

C．单线称文件服务器　　**D．作为有限状态机**

20．与迭代名称解析比较，递归名称解析的优点是（B C）。

A．要求服务器性能高　　**B．缓存结果更为有效**

C．能减少通信开销　　D．算法简单

21．名称用来表示实体，下面属于实体名称的是（BCD）

A．实体图　　**B．标识符**　　**C．易于理解的名称**　　**D．实体地址**

22．下面用于定位移动实体的方法是(ABCD)

A．使用广播与多播　　**B．使用转发指针**

C．给实体指定一个起始位置　　**D．创建一棵分层搜索树**

23．分布式系统的全局状态是指(B D)。

A．用于计算的临时记录　　**B．每个进程的本地状态**

C．要发送的消息　　**D．当前正在传输中的消息**

24．面向消息的中间件模型一般提供（ACD）。

A．持久异步通信　　B. RPC 和 RMI

C．电子邮件　　**D．工作流**

25．在分布式系统中，实现事务的方法是（B C）。

A．创建进程　　**B．为进程分配私有工作空间**

C．做写前日志　　D．创建线程

26．并发控制的总体思想是（ A ）。

A．正确调度相冲突的操作　　B．对事务进行管理

C．进行数据的更新　　D．有序的通信

27．下面属于进程间同步算法的是（C D）。

A．FIFO 算法　　B．基于优先级的算法

C．选举算法　　**D．互斥算法**

28．严格一致性中存在的问题是（A）。

A．依赖于绝对的全局时间　　B．不依赖于绝对的全局时间

C．依赖于并发控制算法　　D．不依赖于并发控制算法

29．下列属于“以数据为中心的一致性模型”是（ACD）。

A．线性化和顺序一致性　　B．最终一致性

C．因果一致性　　**D. FIFO 一致性**

30．下列属于“以客户为中心的一致性模型”是（BCD）。

A．严格一致性　　**B．单调读一致性**

C．写后读一致性　　**D．读后写一致性**

31．下面属于一致性协议的是（C D）。

A．传输协议　　B．中间件协议

C．基于主备份的协议　　**D．复制的写协议**

32．基于主备份的协议是指（A B）

A．负责协调 X 上的远程写操作　　**B．负责协调 X 上的本地写操作**

C．主动复制　　D．协调操作

33．冗余是获得容错性所需的关键技术，下面属于冗余掩盖故障的是（BCD）。

A．存储器冗余　　**B．信息冗余**

C．时间冗余　　**D．物理冗余**

34．在可靠多播通信中，解决反馈拥塞的方法是（A B）。

A．无等级的反馈控制　　**B. 分等级的反馈控制**

C．分层的反馈控制　　D. 闭环反馈控制

35．实现可靠原子多播的方法是（B C）。

A．转发指针　　B. 消息排序

C．虚拟同步　　D. 指针缓存

36．在分布式系统安全设计问题中，控制的焦点是（ABC）。

A．防止无效操作的保护　　**B. 防止未经授权调用的保护**

C．防止未经授权用户的保护　　D. 安全机制的保护

37．在分布式系统中，加密和解密的实现是（D）。

A．递归算法　　B. 安全管理

C．通道管理　　**D. 以密钥为参数的加密算法**

38．分布式系统常用的加密系统有（ABC）。

A．对称加密系统（DES）　　**B. 公钥加密系统（RSA）**

C．散列函数（MDS）　　D. 授权加密

39．在分布式系统安全通道的通信中，其安全性归结为（B C）。

A．授权访问　　B. 对通信各方进行身份验证

C．确保消息完整性和机密性　　D. 检验权限

40．在分布式安全访问控制中，实现访问控制的方式有（A BC）。

A．构造访问控制矩阵　　**B. 构造保护域**

C．采用防火墙　　D. 密钥管理

三．简答题（每小题 n 分，共 m 分）

1．按照资源共享的观念定义的计算机网络具备哪几个主要特征？

答：三个主要特征：1.建立的目的是实现计算机资源的共享，包括数据资源\软件资源和硬件资源。2.互连的计算机是分布在不同地理位置的多台独立的”自治计算机”。3.连网的计算机之间的通信必须遵循共同的网络协议。

2．为什么传输层通信服务常常不适于构建分布式应用程序？

答：因为它不适合用于支持多层客户-服务器交互过程所使用的同步请求-应答方式，在可靠传输中，造成许多开销都耗费在连接的管理上。

3．描述一下客户和服务器之间使用套接字的无连接通信是如何进行的？

答：首先服务器和客户端都要创建一个套接字，并遵循 UDP 协议，服务器将其所在的 IP 地址以及一个端口号绑定到套接字，完成绑定后，服务器就能接收来自客户端的 UDP 数据包了。同样，客户端在创建套接字后，能够向服务器发送 UDP 包进行通信，通信过程中，服务器和客户端之间是不用建立连接的。

4．简述 TCP 和 UDP 协议在通信中的区别

TCP 是面向连接的可靠的协议，适用于传输大批量的文件，检查是否正常传输。而 UDP 是面向非连接的不可靠的协议，适用于传输一次性小批量的文件，不对传输数据报进行检查。

TCP 需要先建立连接才能通话；而 UDP 不需要，实时性要高点。

TCP 可以形象比喻为打电话的过程；UDP 可以比喻为发短信的过程。

TCP 不能发送广播和组播，只能单播；UDP 可以广播和组播。

5．Java RMI 对代码迁移依赖到何种程度？

答：Java RMI 时，每一个方法引用传递实际上就是执行一次代码的迁移，对于移植性差的进程来说，代码迁移是非常必要的。

6．标识符是否可以包含它所引用实体的信息？

答：标识符可以包含它所引用实体的信息，但是，这些信息不允许修改，因为那意味着标识符被改变。

7．在深度为 k 的分层定位服务中，当移动实体改变它的位置时，最多需要更新多少条位置记录？

答：移动实体改变位置会产生删除操作和插入操作，删除操作至少需要更新 k 条位置记录。同样，插入操作也需要更新 k 条位置记录。最后，删除与插入更新移动实体位置的记录共需要 2k+1 条。

8．要使用 Lamport 时间戳实现全序多播，是不是每个消息都必须要被严格地确认？

答：不需要，任何类型的消息，只要它的时间戳大于所接收到的消息的时间戳，就可以被加入消息队列，使用 Lamport 时间戳实现全序多播。

9．许多分布式算法需要使用协调进程。讨论一下，这样的算法实际上可以在什么程度上被看作为分布式的？

答：在集中式算法中，一般会选择一个固定的进程作为协调者，其它的进程可以分布在不同的机器上运行。分布式算法中也同样可以引入协调进程，但是，这个进程并不是固定的，它是从作为算法一部分的进程中选择的。因此，使用协调进程并不会影响算法的分布性。

10．作业调度和进程调度有何区别？

答：作业调度与进程调度之间的差别主要是：作业调度是宏观调度，它所选择的作业只是具有获得处理机的资格，但尚未占有处理机，不能立即在其上实际运行；而进程调度是微观调度，动态地把处理机实际地分配给所选择的进程，使之真正活动起来。另外，进程调度相当频繁，而作业调度执行的次数一般很少。

11．请解释 DNS 如何进行复制，以及它实际运行很好的原因。

答：DNS 进行复制的基本思想是：域名服务器可以缓存以前查找过的结果。由于 DNS 的名称到地址的映射很少更改，因此，这些结果可以缓存很长一段时间。

12．简述进程与程序的联系和区别

答：（1）联系：一个进程可以涉及到一个或几个程序的执行；一个程序可以对应一个或多个进程，即同一程序段可以在不同数据集合上运行，可构成不同的进程，例如打印输出程序段，例如同一高级语言编译程序与多个用户源程序。

（2）进程和程序的区别主要体现在：

1）进程是动态的，具有一定的生命周期，而程序是静态的；

2）进程可并发执行，而没有创建进程的程序是不能执行的；

3）进程是操作系统中申请和分配资源的基本单位，而没有创建进程的程序是不能申请资源的；

4）进程包括程序、数据和进程控制块；

5）同一程序的多次执行对应多个进程

13．在下图中，一个顺序一致的存储器允许 6 种可能的语句交叉。请列举出这 6 种可能的情况。

进程 P1	进程 P2
x=1;	y =1；
if(y== 0) kill(P2)	if(x== 0) kill(P1)

答：这 6 种可能的情况是：

(1) a=1;　if (b= =0);　b=1;　if (a= =0);

(2) a=1;　b=1;　if (a= =0);　if (b= =0);

(3) a=1;　b=1;　if (b= =0);　if (a= =0);

(4) b=1;　if (a= =0);　a=1;　if (b= =0);

(5) b=1;　a=1;　if (b= =0);　if (a= =0);

(6) b=1;　a=1;　if (a= =0);　if (b= =0);

14．一个文件被复制在 10 个服务器上，请列出表决算法允许的所有读团体和写团体。

答：下列可能性的读团体和写团体是合法的：

（1，10）、（2，9）、（3，8）、（4，7）、（5，6）、（6，5）、（7，4）、（8，3）、（9，2）、（10，1）。

15．原子多播的可扩展性重要到哪种程度上？

答：它取决于一组包含多个进程的状态。如果进程为故障容错进行了复制，拥有少量的副本可能就足够了，在这种情况下，可扩展性几乎不成问题。如果是由不同进程构成的组，可扩展性就可能成了一个问题。当为了性能而复制时，原子多播自身可能超出负荷的能力。

16．在两阶段提交协议中，为什么即使在参与者们选择一个新的协调者的情况下也不会完全消除阻塞？

答：因为选举结束后，新的协调者也同样可能会崩溃。在这种情况下，其余的参与者也不能做出最后决定，因为这需要由新当选的协调者发起选举。

17．假设 Alice 希望向 Bob 发送一条消息 m。她没有使用 Bob 的公钥 K^{+}_{B} 加密 m，而是生成了一个会话密钥 $K_{A,B}$，然后发送$[K_{A,B}(m), K^{+}_{B}(K_{A,B})]$。为什么一般来讲，这种方法更好？（提示：考虑性能问题）。

答：会话密钥有一个短而固定的长度，而消息 m 可能是任意长度。因此，采用会话密钥和公钥结合加密短消息通常在性能方面优于只使用一个公钥加密的消息。

18．列举出为密钥管理使用集中式服务的一些优点和缺点。

答：一个显著的优点是简单。比如：若有 N 个客户在一个集中式的服务器上共享了 1 个密钥，我们就只需要维护 N 个密钥；如果是成对共享密钥，那我们就需要维护 N(N-1)/2 个。而且使用集中式服务器存储和维护都在一个站点上，使存储和维护都比较方便。潜在的缺点：首先是服务器有可能成为性能和可用性的瓶颈。其次，如果服务器机密被泄露，就必须建立新的密钥。

19．一个网络中，DNS 服务器应该部署在什么地方最合适？

答：要用域名访问 Internet 上的服务器必须先访问 DNS 服务器，经过 DNS 对域名的解析才能连接到相应的主机。所以，在一个网络中，DNS 服务器应该部署在客户端可以集中访问的网络位置上。

20．进程间同步和互斥的含义是什么？

答：进程间同步是并发进程之间存在的相互制约和相互依赖的关系。

进程间互斥是若干进程共享一资源时，任何时刻只允许一个进程使用。

四．综合题（本题结果不是唯一的，每小题 n 分，共 m 分）

1．有三个进程 P1，P2 和 P3 并发工作。进程 P1 需用资源 S3 和 S1；进程 P2 需用资源 S1 和 S2；进程 P3 需用资源 S2 和 S3。回答：

(1)若对资源分配不加限制，会发生什么情况?为什么？

(2)为保证进程正确工作，应采用怎样的资源分配策略?为什么？

(1)多个进程动态地共享系统的资源可能会产生死锁现象。死锁的产生，必须同时满足四个条件，第一个是互斥条件，即一个资源每次只能由一个进程占用；第二个为等待条件，即一个进程请求

资源不能满足时，它必须等待，但它仍继续保持已得到的所有其它资源；第三个是非出让条件，任何一个进程不能抢占另一个进程已经获得且未释放的资源；第四个为循环等待条件，系统中存在若干个循环等待的进程，即其中每一个进程分别等待它前一个进程所持有的资源。防止死锁的机构只须确保上述四个条件之一不出现，则系统就不会发生死锁。

只要资源分配策略能保证进程不出现循环等待，则系统就不会发生死锁。

(2) 银行家算法分配资源的原则是：系统掌握每个进程对资源的最大需求量，当进程要求申请资源时，系统就测试该进程尚需资源的最大量，如果系统中现存的资源数大于或等于该进程尚需的最大量时，则就满足进程的当前申请。这样可以保证至少有一个进程可能得到全部资源而执行到结束，然后归还它所占用的全部资源供其它进程使用。银行家算法破坏了产生死锁的第四个条件，即不可能产生循环等待，从而可以避免死锁的发生。

防止进程发生循环等待的另一种资源分配策略是按序分配算法，其基本思想如下：把系统中所有的资源排一个顺序，例如系统共有 m 个资源，用 ri 表示第 i 个资源，那么这 m 个资源是：

r1, r2, r3 ……, rm

规定任何进程不得在占用资源 ri（1<i≤m）后再申请 rj（j<i≤m），或者说，如果里程需要资源 rj，那么它必须在申请 ri 之前申请（j<i）。可以证明，按这种策略分配资源时破坏了循环等待条件，故能防止发生死锁

2．如何设计一个好的 RPC？采用 client/server 模型与应用程序的组件说明开发过程？

本文比较详细地介绍了远程过程调用(RPC)的 OSF 标准在 Microsoft VC++中的实现原理, 以及如何使用它们来开发应用程序. 阅读本文你将了解 RPC 的基本原理, 并将看到如何开发使用 RPC 进行异种机网络分布式处理的客户机应用程序和服务器应用程序.

Para 1. RPC 工作原理

RPC 是把传统本地过程调用的概念加以扩充后引入分布式环境的一种形式. RPC 的形式和行为与传统本地过程调用极为相似, 差别仅在于被调用的 procedure(过程)实际运行在与调用者的场点不同的场点上(如图 1). 也正是由于这一差别, 我们得通过编写程序来实现两场地之间的连接和信息沟通.

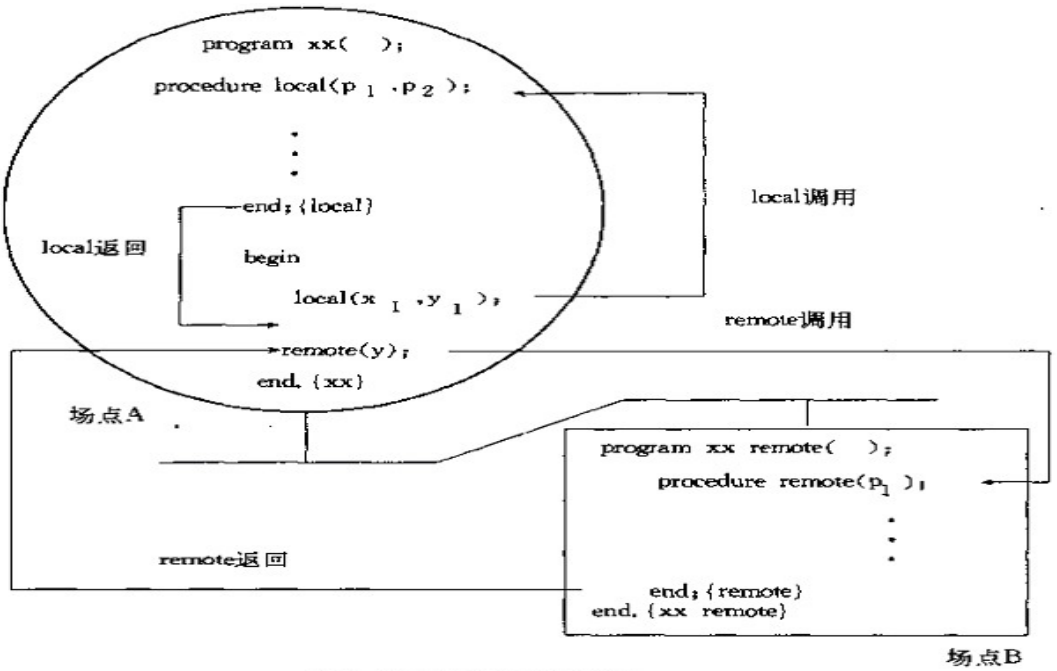


图 1 远程过程调用示意图

RPC 机制的实质是实现 OSI 七层模型中的会话层功能. 它在两个试图进行通信的场点之间建立一条逻辑信道(即会话连接), 并利用这个信道交换信息, 不用时就释放连接. 下面我们就来看看 RPC 的通信模型(如下图 2):

Client 端:

1) 发送远程过程调用的消息(以消息包形式)给远程的 server 端;
2) 等待, 直到收到 server 端对该请求的回复;
3) 一旦接收到来自 server 端的返回执行结果, 就继续执行后面的程序.

Server 端:

1) 倾听状态, 等待 client 端发送过程调用消息;
2) 一旦接收到过程调用消息, server 就抽取参数并分析它, 然后执行所请求的过程;
3) 将执行结果以消息包形式回送给 client.

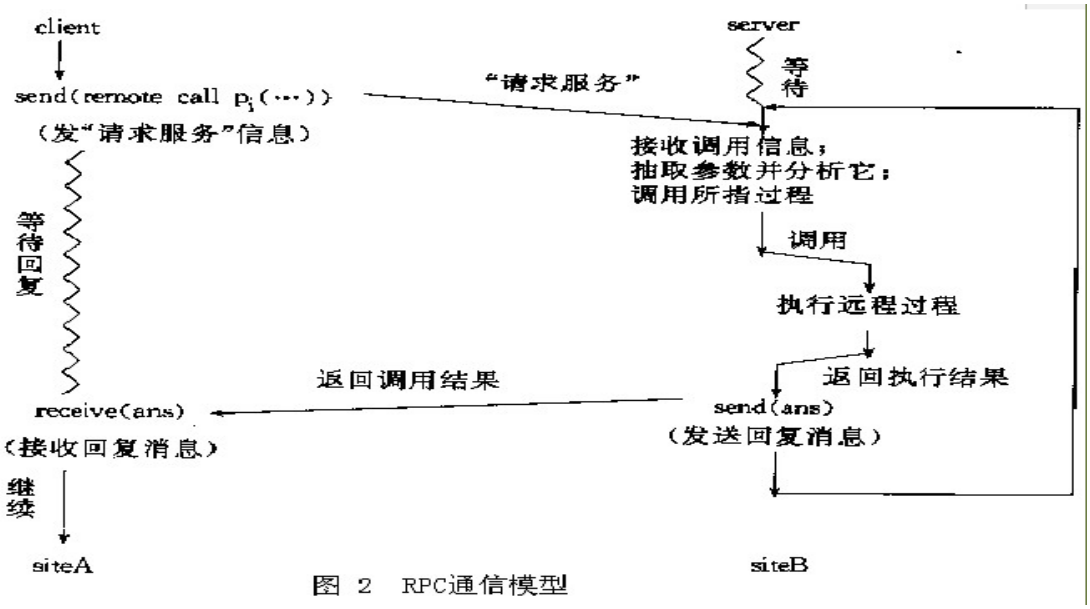


图 2 RPC通信模型

至于 RPC 的具体实现, 我们可以借助下图 3 来理解:

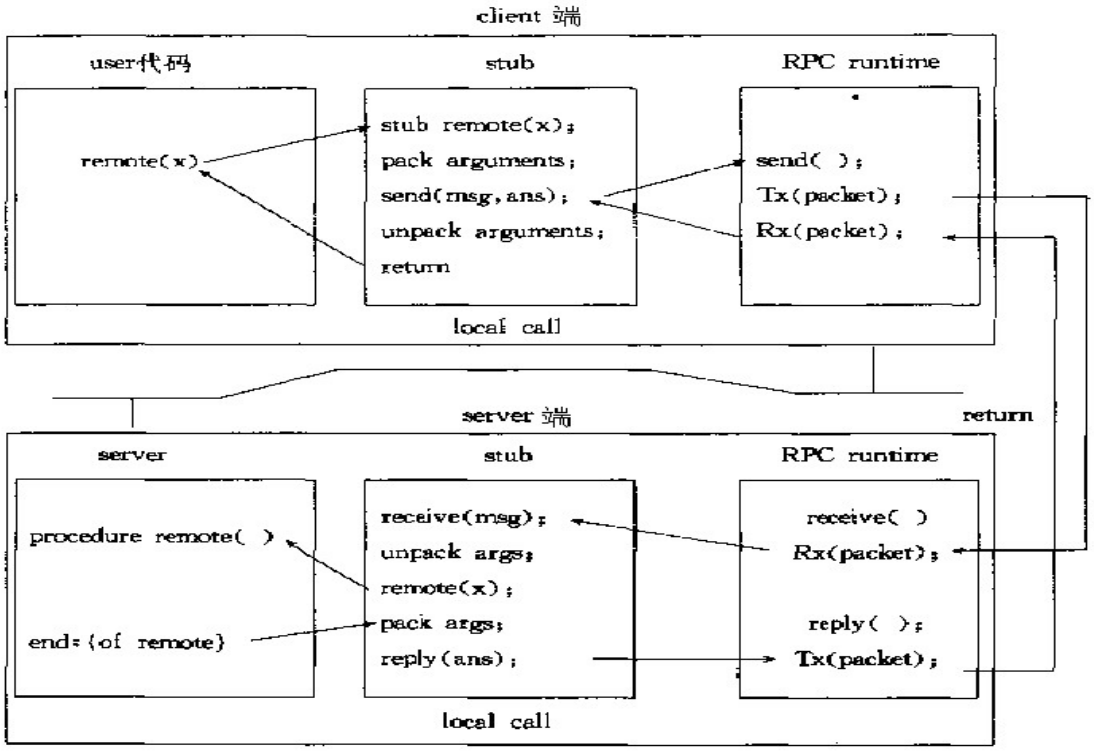


图 3 RPC 的实现概况

其中 stub 是一组 RPC 机制的操作原语，这些原语构成了 RPC 的实现细节，它可以独立于 client、server 编程. 下面我们来解说图 3 的执行过程:

1) 调用者调用本地 stub 中的一个过程(开始远程过程调用请求).

2) 这个 stub 过程把有关的参数组装成一个消息包或一组消息包，形成一条消息. 运行此执行过程的远程场点的 IP 地址和执行该过程的进程 ID 号也包含在这条消息中.

3) 将这条消息发送给对应的 RPC runtime(RPC 运行库)子程序，由这个子程序将消息发送到远程场点.

4) 在接收到这条消息时，server 端的 RPC runtime 子程序引用与被调用者对应的 stub 中的一个子程序，并让它来处理消息.

5) 与被调用者对应的 stub 中的这个子程序撤卸消息，解析出相关参数，并用本地调用方式执行所指定的过程.

6) 返回调用结果，调用者对应的 stub 子程序执行 return 语句返回到用户，整个 RPC 过程结束.

实际上，从上面这个执行过程中，我们可以看到 RPC 的实现主要有两个问题需要解决. 一个是在远程过程调用时，如何定位远程场点；另外一个就是相关的两个场点必须能协同工作，所有这些工作对用户都是透明的，依次执行.

通常在实际编程中，程序设计者主要负责设计计算过程并实现计算过程体，而对应的 stub 由系统生成. 后面我们就要说到 Microsoft 的 RPC 实现机制，看看它是如何产生 stub 的。

Para2. 如何设计好的 RPC

对调用双方来说，传递 RPC 参数包括辅助处理本地数据表示和网络数据表示的相互转换. 此外，输入输出参数需要一些存储分配. 同时，RPC 中的等待时间也不能忽略.

所以，一般来说，应该尽可能降低调用次数. 例如，如果要对一个大数组的每个元素都执行计算，我们就可以一次调用处理一整行或者整个数组，而不用每次调用传递一个元素. 这样可以降低有 RPC 引入的额外开销.

Para3. 应用程序的组件
为了在 client 和 server 端使用 RPC, 当然少不了 client 进程和 server 进程. 另外还有一个名称服务进程, 这个稍后再说.
开发过程大致是这样的:
1) 任何 RPC 调用都使用一个定义在 IDL(interface definition language, 接口定义语言)文件中的接口, 然后 MIDL(Microsoft IDL)编译器对 IDL 文件进行编译, 编译之后会自动生成一个.h 文件, 同时生成一个 client stub 和一个 server stub. 关于这个你可以在 dos 下运行 midl.exe/?得到更详细的信息.
2) Client 端应用程序使用 client stub 调用 RPC runtime 以实现网络上的调用. 接下来 RPC 运行时使用一组 DLL 中的一个来实现被使用的特定网络协议.
3) Server 端也与 RPC runtime 连接. 不过 server 端应用程序使用一些其它的函数来将自己作为一个特殊接口的服务器进行注册(向谁注册?), 并开始侦听接口的请求.

3. 如何用 Java 或其它语言解决线程同步与互斥的问题？(任选择一种语言)

同步 有同步方法和同步块
当有多个线程的时候，经常需要去同步这些线程以访问同一个数据或资源。例如，假设有一个程序，其中一个线程用于把文件读到内存，而另一个线程用于统计文件中的字符数。由于每个操作都有自己的线程，操作系统会把两个线程当作是互不相干的任务分别执行，这样就可能在没有把整个文件装入内存时统计字数。为解决此问题，你必须使两个线程同步工作。存在一些线程同步地址的问题，Win32 提供了许多线程同步的方式。在本节你将看到使用临界区、互斥、信号量和事件来解决线程同步的问题。
1. 临界区
临界区是一种最直接的线程同步方式。所谓临界区,就是一次只能由一个线程来执行的一段代码。如果把初始化数组的代码放在临界区内，另一个线程在第一个线程处理完之前是不会被执行的。在使用临界区之前，必须使用 InitializeCriticalSection()过程来初始化它。
其声明如下：

```
procedure InitializeCriticalSection(var lpCriticalSection:
  TRLCriticalSection);stdcall;
```

lpCriticalSection 参数是一个 TRTLCriticalSection 类型的记录，并且是变参。只需要在 lpCriticalSection 中传递未初始化的记录，InitializeCriticalSection()过程就会填充这个记录。
在记录被填充后，我们就可以开始创建临界区了。这时我们需要用 EnterCriticalSection()和 LeaveCriticalSection()来封装代码块。这两个过程的声明如下：

```
procedure EnterCriticalSection(var lpCriticalSection:TRRLCriticalSection);stdcall;
procedure LeaveCriticalSection(var lpCriticalSection:TRRLCriticalSection);stdcall;
```

正如你所想的，参数 lpCriticalSection 就是由 InitializeCriticalSection()填充的记录。
当你不需要 TRTLCriticalSection 记录时，应当调用 DeleteCriticalSection()过程，下面是它的声明：

```
procedure DeleteCriticalSection(var lpCriticalSection: TRTLCriticalSection);
stdcall;
```

2. 互斥
互斥非常类似于临界区，除了两个关键的区别：首先，互斥可用于跨进程的线程同步。其次，互斥能被赋予一个字符串名字，并且通过引用此名字创建现有互斥对象的附加句柄。
提示临界区与事件对象(比如互斥对象)的最大的区别是在性能上。临界区在没有线程冲突时，要用１０～１５个时间片，而事件对象由于涉及到系统内核要用 400~600 个时间片。可以调用函数

CreateMutex ( )来创建一个互斥量。下面是函数的声明：

```
function CreateMutext(lpMutextAtrribes:PSecurityAttributtes;
  bInitalOwner:BOOL; lpName:PChar):THandle; stdcall;
```

lpMutexAttributes 参数为一个指向 TSecurityAttributtes 记录的指针。此参数通常设为 0，表示默认的安全属性。bInitalOwner 参数表示创建互斥对象的线程是否要成为此互斥对象的拥有者。当此参数为 False 时，表示互斥对象没有拥有者。lpName 参数指定互斥对象的名称。设为 nil 表示无命名，如果参数不是设为 nil，函数会搜索是否有同名的互斥对象存在。如果有，函数就会返回同名互斥对象的句柄。否则，就新创建一个互斥对象并返回其句柄。当使用完互斥对象时，应当调用 CloseHandle()来关闭它。

在程序中使用 WaitForSingleObject()来防止其他线程进入同步区域的代码。此函数声明如下：

```
function WaitForSingleObject(hHandle: THandle; dwMilliseconds: DWORD): DWORD; stdcall;
```

这个函数可以使当前线程在 dwMilliseconds 指定的时间内睡眠，直到 hHandle 参数指定的对象进入发信号状态为止。一个互斥对象不再被线程拥有时，它就进入发信号状态。当一个进程要终止时，它就进入发信号状态。dwMilliseconds 参数可以设为 0，这意味着只检查 hHandle 参数指定的对象是否处于发信号状态，而后立即返回。dwMilliseconds 参数设为 INFINITE，表示如果信号不出现将一直等下去。再次声明，当一个互斥对象不再被一个线程所拥有,它就处于发信号状态。此时首先调用 WaitForSingleObject()函数
的线程就成为该互斥对象的拥有者，此互斥对象设为不发信号状态。当线程调用 ReleaseMutex()函数并传递一个互斥对象的句柄作为参数时，这种拥有关系就被解除，互斥对象重新进入发信号状态。

注意除 WaitForSingleObject()函数外，你还可以使用 WaitForMultipleObject()和
MsgWaitForMultipleObject()函数，
它们可以等待几个对象变为发信号状态。这两个函数的详细情况请看 Win32 API 联机文档。

3. 信号量

另一种使线程同步的技术是使用信号量对象。它是在互斥的基础上建立的，但信号量增加了资源计数的功能，预定数目的线程允许同时进入要同步的代码。可以用 CreateSemaphore()来创建一个信号量对象，其声明如下：

```
  function CreateSemaphore(lpSemaphoreAttributes: PSecurityAttributes;
    lInitialCount, lMaximumCount: Longint; lpName: PChar): THandle; stdcall;
```

和 CreateMutex()函数一样，CreateSemaphore()的第一个参数也是一个指向
TSecurityAttribute s 记录的指针，此参数的缺省值可以设为 nil。

lInitialCount 参数用来指定一个信号量的初始计数值,这个值必须在 0 和 lMaximumCount 之间。此参数大于 0,就表示信号量处于发信号状态。当调用 WaitForSingleObject()函数(或其他函数)时，此计数值就减 1。当调用 ReleaseSemaphore()时，此计数值加 1。

参数 lMaximumCount 指定计数值的最大值。如果这个信号量代表某种资源，那么这个值代表可用资源总数。

参数 lpName 用于给出信号量对象的名称，它类似于 CreateMutex()函数的 lpName 参数。

4．如何用 Java 或 C/C++语言实现多线程？(任选择一种语言)

```
//这个例子是基于事件对象的
#include <windows.h>
#include <iostream.h>

DWORD WINAPI Fun1Proc(LPVOID lpParameter);//thread data
DWORD WINAPI Fun2Proc(LPVOID lpParameter);//thread data
```

```
int tickets=100;
HANDLE g_hEvent;
void main()
{
	HANDLE hThread1;
	HANDLE hThread2;
	//创建人工重置事件内核对象
	g_hEvent=CreateEvent(NULL,FALSE,FALSE,"tickets");
	if (g_hEvent)
	{
		if (ERROR_ALREADY_EXISTS==GetLastError())
		{
			cout<<"only one instance can run!"<<endl;
			return;
		}
	}
	SetEvent(g_hEvent);

	//创建线程
	hThread1=CreateThread(NULL,0,Fun1Proc,NULL,0,NULL);
	hThread2=CreateThread(NULL,0,Fun2Proc,NULL,0,NULL);
	CloseHandle(hThread1);
	CloseHandle(hThread2);

	//让主线程睡眠 4 秒
	Sleep(4000);
	//关闭事件对象句柄
	CloseHandle(g_hEvent);
}
//线程 1 的入口函数
DWORD WINAPI Fun1Proc(LPVOID lpParameter)//thread data
{
	while (true)
	{
		WaitForSingleObject(g_hEvent,INFINITE);
		//ResetEvent(g_hEvent);
		if (tickets>0)
		{
			Sleep(1);
			cout<<"thread1 sell ticket :"<<tickets--<<endl;
			SetEvent(g_hEvent);
		}
		else
```

```
            {
                SetEvent(g_hEvent);
                break;
            }
        }

        return 0;
    }
    //线程 2 的入口函数
    DWORD WINAPI Fun2Proc(LPVOID lpParameter)//thread data
    {
        while (true)
        {
            //请求事件对象
            WaitForSingleObject(g_hEvent,INFINITE);
            //ResetEvent(g_hEvent);
            if (tickets>0)
            {
                Sleep(1);
                cout<<"thread2 sell ticket :"<<tickets--<<endl;
                SetEvent(g_hEvent);
            }
            else
            {
                SetEvent(g_hEvent);
                break;
            }
        }

        return 0;
    }
```

http://topic.csdn.net/u/20080728/13/dd41d46a-6224-4aaf-b484-2e21e181c8e5.html

5．分布式令牌环算法存在令牌丢失的问题，如果令牌丢失，会导致算法失败，请将该算法改进一下，使该算法既能检测到令牌丢失，也能进行补救。

令牌环的维护

令牌环的故障处理功能主要体现在对令牌和数据帧的维护上。令牌本身就是比特串，绕环传递过程中也可能受干扰而出错，以至造成环路上无令牌循环的差错；另外，当某站点发送数据帧后，由于故障而无法将所发的数据帧从网上撤消时，又会造成网上数据帧持续循环的差错。令牌丢失和数据帧无法撤消，是环网上最严重的两种差错，可以通过在环路上指定一个站点作为主动令牌管理站，以此来解决这些问题。

主动令牌管理站通过一种超过机制来检测令牌丢失的情况，该超时值比最长的帧为完全遍历环路所需的时间还要长一些。如果在该时段内没有检测到令牌，便认为令牌已经丢失，管理站将清除环路上的数据碎片，并发出一个令牌。

为了检测到一个持续循环的数据帧，管理站在经过的任何一个数据帧上置其监控位为 1，如

果管理站检测到一个经过的数据帧的监控拉的已经置为 1，便知道有某个站未能清除自己发出的数据帧，管理站将清除环路的残余数据，并发出一个令牌。

6．散列函数为什么是安全的？散列函数的基本要求有哪些？常用散列函数的构造方法有哪些？

（1）Hash 函数就是把任意长的输入消息串变化成固定长的输出串的一种函数。这个输出串称为该消息的杂凑值。Hash 函数输出并不以可辨别的方式依赖于输入;在任何输入串中单个比特的变化,将会导致输出比特串中大约一半的比特发生变化。所以散列函数是安全的。一般用于产生消息摘要，密钥加密等.

（2）一个安全的散列函数应该至少满足以下几个条件:

①输入长度是任意的;

②输出长度是固定的,根据目前的计算技术应至少取 128bits 长,以便抵抗攻击;

③对每一个给定的输入,计算输出是很容易的

④给定散列函数的描述,找到两个不同的输入消息杂凑到同一个值是计算上不可行的,或给定散列函数的描述和一个随机选择的消息,找到另一个与该消息不同的消息使得它们杂凑到同一个值是计算上不可行的。

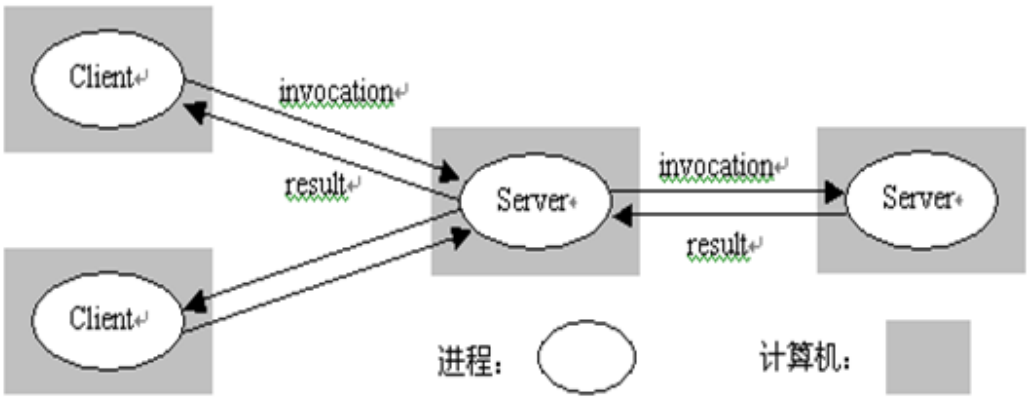
（3）散列函数的选择有两条标准：简单和均匀。常用散列函数的构造方法有：平方取中法、除余法、相乘取整法、随机数法。

常见散列函数(Hash 函数)：•MD5 •SHA•MAC •CRC

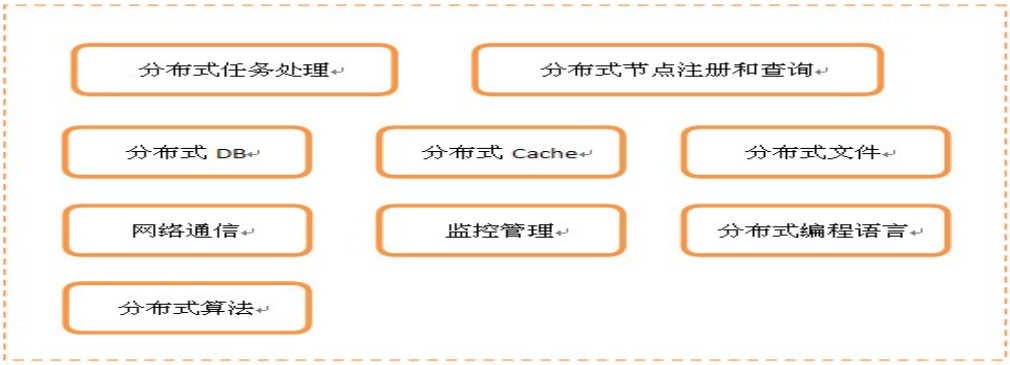
7．分布式可繁也可以简，请你组建一个最简单的分布式系统模型。

Client-server模型

这是分布式系统最经常被引用的体系结构，下图阐述了这种简单的结构：客户进程与各服务器进程（在分离的主计算机上）交互以访问它们所管理的共享资源。



8．一个最完备的分布式体系由以下模块组成。请说明各模块的功能？



①分布式处理系统必须有能力在短时间内动态地组合成面向不同服务对象的系统。对用户来说系统是透明的，用户只需指定系统干什么而不必指出哪个部件可以提供这一服务。系统各组成部分是自主的，但不是无政府状态，而是遵循某个主计划由高级操作系统进行协调工作。在一个计算机网中有多台主机不一定都是分布式处理。如果这样的系统不具备动态组合及任务再指派的能力，那么它们仍然是集中式处理。

②分布式查询可以访问来自多种异类数据源的数据，而这些数据可存储在相同或不同的计算机上。

③分布式数据库系统由分布于多个计算机结点上的若干个数据库系统组成,它提供有效的存取手段来操纵这些结点上的子数据库。分布式数据库在使用上可视为一个完整的数据库,而实际上它是分布在地理分散的各个结点上。当然,分布在各个结点上的子数据库在逻辑上是相关的。

④分布式缓存支持一些基本配置：重复（replicated）、分配（partitioned）和分层（tiered）。重复（Replication）用于提高缓存数据的可用性。在这种情况下，数据将重复缓存在分布式系统的多台成员机器上，这样只要有一个成员发生故障，其他成员便可以继续处理该数据的提供。另一方面，分配（Partitioning）是一种用于实现高可伸缩性的技巧。通过将数据分配存放在许多机器上，内存缓存的大小加随着机器的增加而呈线性增长。结合分配和重复这两种机制创建出的缓存可同时具备大容量和高可伸缩的特性。

⑤分布式文件系统具有执行远程文件存取的能力,并以透明方式对分布在网络上的文件进行管理和存取。

⑥ 分布式网络通信能够连接跨越了多台计算机的应用程序各节点。

⑦分布式监控管理可以有效避免最上层服务器因顾及不暇而出现管理疏漏的现象。

⑧用于编写运行于分布式计算机系统上的分布式程序。一个分布式程序由若干个可以独立执行的程序模块组成,它们分布于一个分布式处理系统的多台计算机上被同时执行。它与集中式的程序设计语言相比有三个特点：分布性、通信性和稳健性。

⑨分布式系统的执行存在着许多非稳定性的因素。分布式算法起到分布性和并发性的作用，这一点不同于集中式算法。

9．设计一个分布式网络管理系统的架构与开发模型。（200 字左右）

分布式网络管理系统的实现主要有对等式、层次式和混合式三种实现方式。

对等式（P2P）网络管理：网管功能被分布到多个管理者上，完成各自域内的网络逻辑管理（综合管理），而每个被管设备都是具有一定自我管理能力的自治单元。

层次式网络管理：引入中层管理站 MLM（Middle-Level Manager）以减轻顶层管理站 MOM（Manager Of Managers）的负担，减少网络传输、消除瓶颈，增加可靠性和扩展性，从而提高整个网络管理系统的性能。是一种很具生命力的方法。

混合式网络管理：它结合了两者的优点，但当网络规模扩大时，集成管理站和单元管理站的增多将导致管理关系复杂性的非线性增长。

请围绕分布式管理的主题，选择一种方式，依次从以下三个方面论述。
（1）结合你参与或你熟悉网络管理架构一个简单的模型；
（2）简要说明数据一致性算法；
（3）简述采用的数据安全性要求。
对等式（P2P）网络管理：
（1）把整个网络分为若干个对等的域，每个域设置一个网络管理者，负责集中管理域中的设备。因此．在一个网络系统中可以有多个管者．几个对等的管理者可以同时在网络系统中运行。每个管理者负责管理网络系统中的一个特定域．管理者之间可以相互通信，或者通过高层管理者进行协调。如下图 2 所示．

基于 P2P 的分布式网络管理模型主要由 3 部分组成：被管设备，网管工作站（网管节点），网管中心，如图 1 所示。

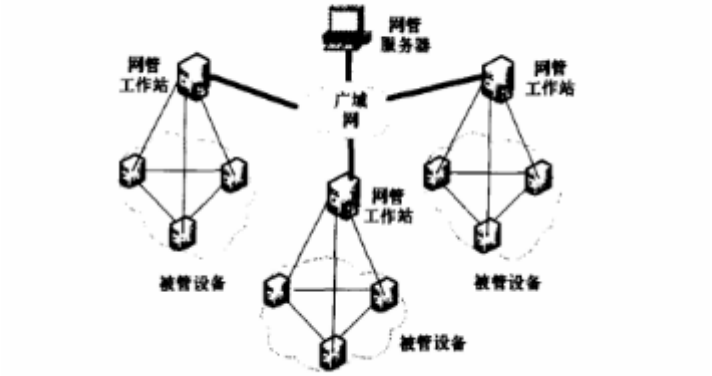


图 1 基于 P2P 的分布式网管模型

管理信息库似(MIB)
（2）P2P 系统中基于副本链的一致性维护算法：一种无结构纯 P2P 的副本一致性维护算法。利用副本节点发起的第一次更新消息在 P2P 网络中的广播，由其他收到消息的节点给出响应，构建副本链。副本链建立后，更新消息在副本节点间进行传播，不再在网络中洪泛。
（3）P2P 环境对数据安全性要求一般，通常是利用 UDP 来交换数据，而不是 TCP String 套接字。

10．论分布式共享存储一致性协议的关键技术（200 字左右）。

分布式共享存储可以使那些原来彼此独立的计算机共享一个统一的地址空间,称作虚拟共享存储空间。分布式共享存储层必需负责维护一致性。也就是说，任何处理机的读操作都要保证返回最新写的值，而不管这个写操作是由谁来执行的。一般来说，虚拟共享存储空间是按页进行管理的，当对一个远程的共享数据进行访问时，将会在本产生一个缺页，这个缺页将被软件共享存储层截获并负责从相应的处理机将需要的共享数据取来，并进行相应的一致性维护。如果这个共享数据在其他处理机上有副本，那么还要与其他处理机通信来维护一致性。高速缓存的一致性设计为其关键技术。典型：基于目录的一致性协议，包括目录状态设置及其转换、转发策略、死锁处理机制等。
新型：基于锁的高速缓存一致性协议。