2013年

一

联系：一般的分布式系统是建立在计算机网络之上的，因此分布式系统与计算机网络在物理结构上基本相同；计算机网络为分布式系统研究提供了技术基础，而分布式系统是计算机网络技术发展的高级阶段。

主要区别：网络与分布式系统的主要区别在于软件（特别是操作系统）而不是硬件。分布式系统在用户看来只是一个模型，在操作系统之上有一层软件（中间件）来负责实现，强调了系统的一致性和透明性，使得用户感觉像是在使用一台电脑；而计算机网络更强调网络的组织，而不考虑一致性等问题，用户看到的是实际的机器，系统并没有使这些机器看起来一样或试图保持一致，机器之间的差异对于用户来说是完全可知的。

网络操作系统是运行在松耦合硬件上的松耦合的软件。除了共享文件系统外，对用户显而易见的是这个系统包含有许多计算机。每台计算机都运行它自己的操作系统。除了客户/服务器通信必须遵守协议之外，计算机之间基本上没有什么合作。

分布式系统是在网络计算机的集合上运行，但行为却像一个虚拟的单处理机，用户意识不到系统中由多个CPU存在，这种性质称为单一系统映像。目前还没有一个系统能完全满足这一要求。

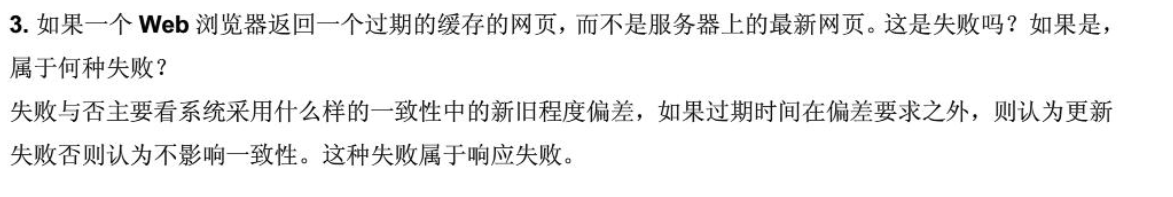
二

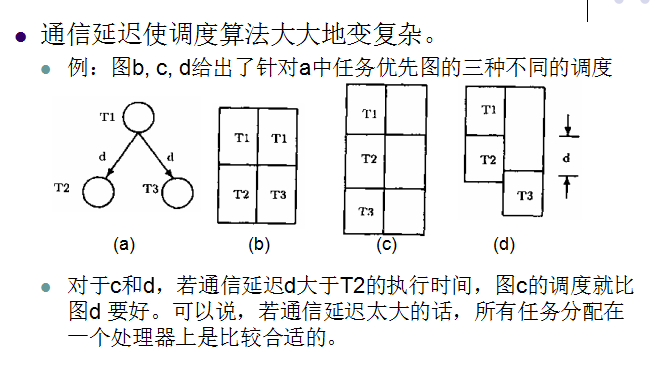
答：二者并不相同。异步RPC向调用者返回一个通知，这意味着客户第一次调用之后，有一个额外的消息会被发送。类似地，服务端接收到它的响应消息已经发送到客户端的通知。如果保证通信可靠的话，两次异步RPC调用是一样的。

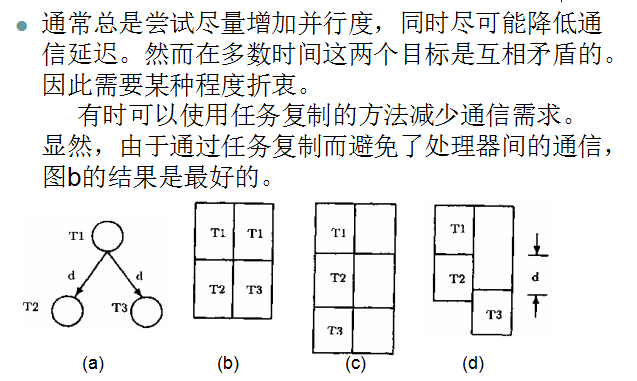
三

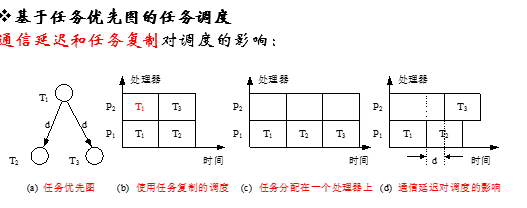
答： 不能。如果更新进程接收到它的更新请求已被处理的通知，它就从数据存储断开并重新连接到另一个副本。更新是否应用在副本上则没有通知。相比来说，如果使用一个阻塞协议，更新进程仅在他的更新请求被完全应用在其它副本上时才断开连接。

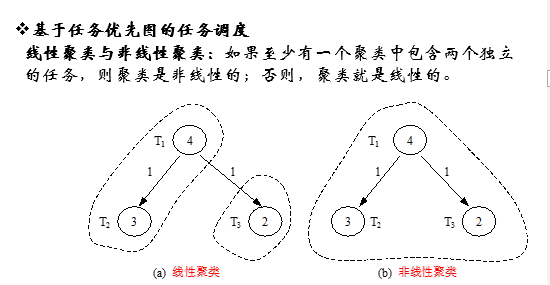
四

五

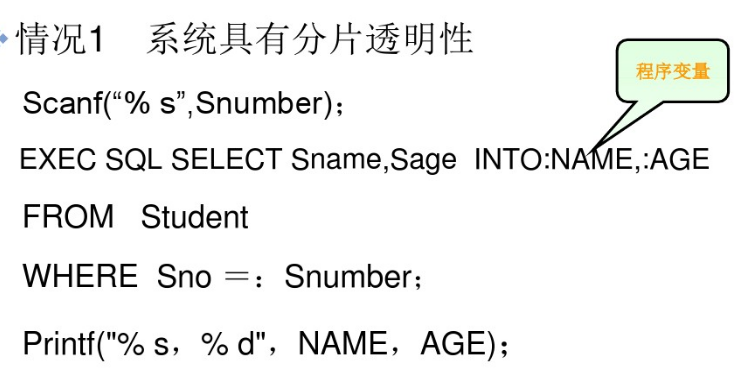


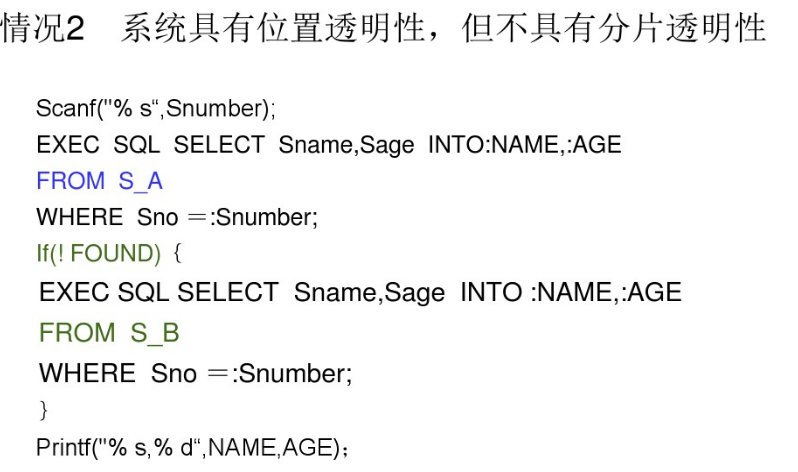


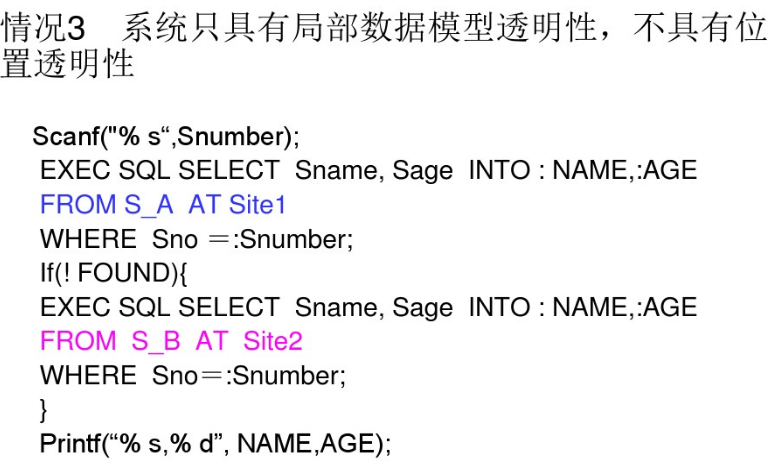




六







2014年

一

联系：

一般的分布式系统是建立在计算机网络之上的，因此分布式系统与计算机网络在物理结构上基本相同；计算机网络为分布式系统研究提供了技术基础，而分布式系统是计算机网络技术发展的高级阶段。

主要区别：

网络与分布式系统的主要区别在于软件（特别是操作系统）而不是硬件。

分布式系统在用户看来只是一个模型，在操作系统之上有一层软件（中间件）来负责实现，强调了系统的一致性和透明性，使得用户感觉像是在使用一台电脑；

而计算机网络更强调网络的组织，而不考虑一致性等问题，用户看到的是实际的机器，系统并没有使这些机器看起来一样或试图保持一致，机器之间的差异对于用户来说是完全可知的。

二

答：令k为服务器的数量，则有(1/4)k<0.01。即在最坏的情况下，这时所有的服务器都已关闭，发生这种情况的概率是1/100。因此k=4。

三

答：它们通常不提供分布透明性，这意味着应用程序开发人员需要注意通信的实现，从而导致解决方案的可扩展性很差。分布式应用程序，例如基于套接字构建的分布式应用程序，将很难移植或者和其它应用程序交互。

四

答：当一个进程接受到ELECTION消息，它会检查消息是谁开始发送的，如果是它自己开始的（例如它的位置在列表的首位），它会把消息变成协调者消息并在正文里描述它。如果消息不是它开始的，它会加入其进程号并沿着环向前发送。然而，如果它更早地发送了自己的选举消息并发现了竞争者，它会将创始人的进程号与自己的相比较。如果其他进程拥有较小的号码，它会丢弃那个消息而不是传递消息。如果竞争者更大，那么消息将以平常的方式发送。这样，如果多个选举消息被开始发送，那么入口最大的消息将会幸存，其他消息将沿着路由被忽略掉。

五

没找到

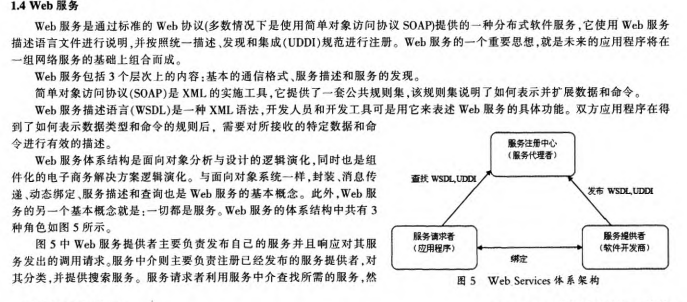
2015年

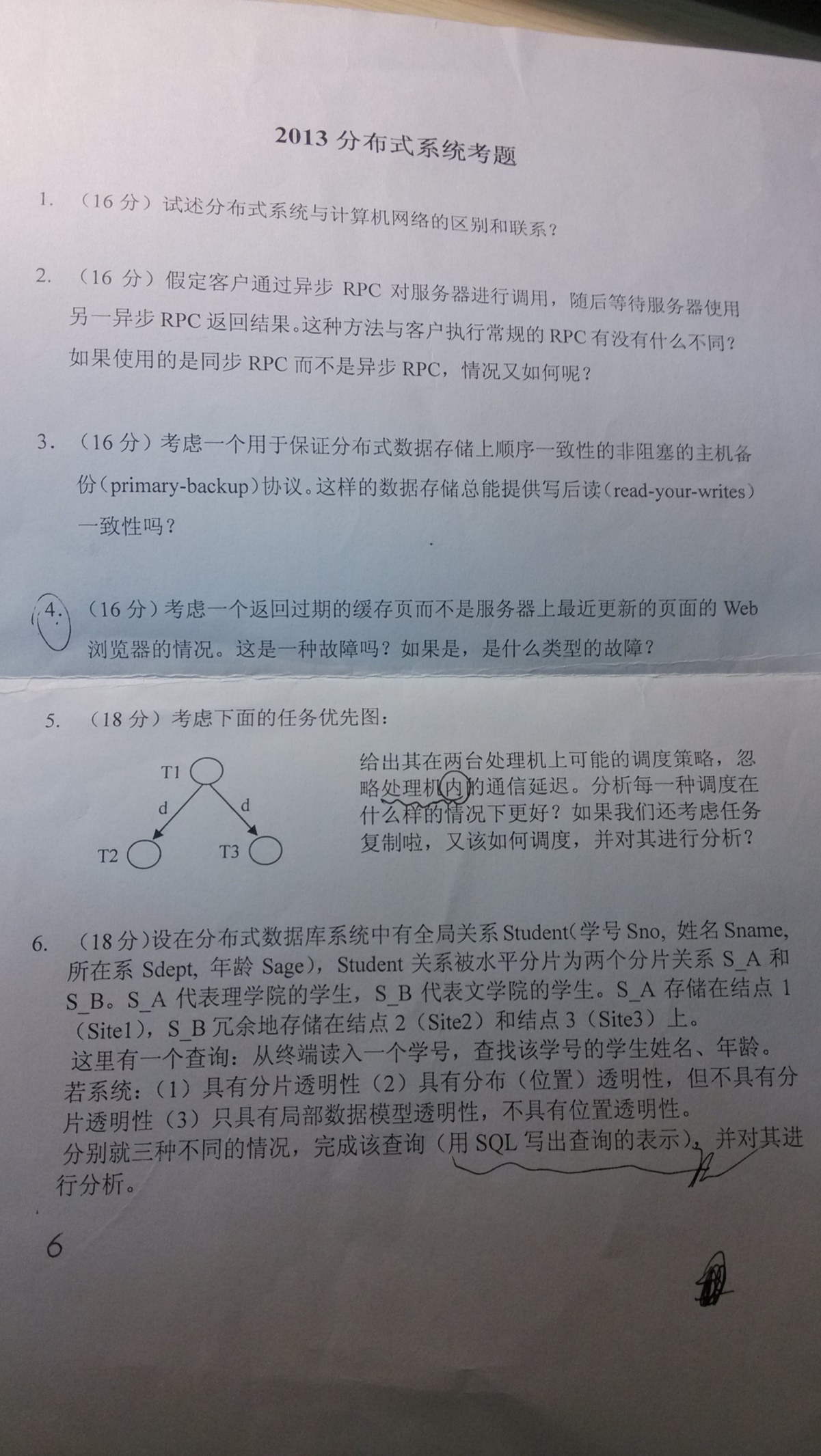
二

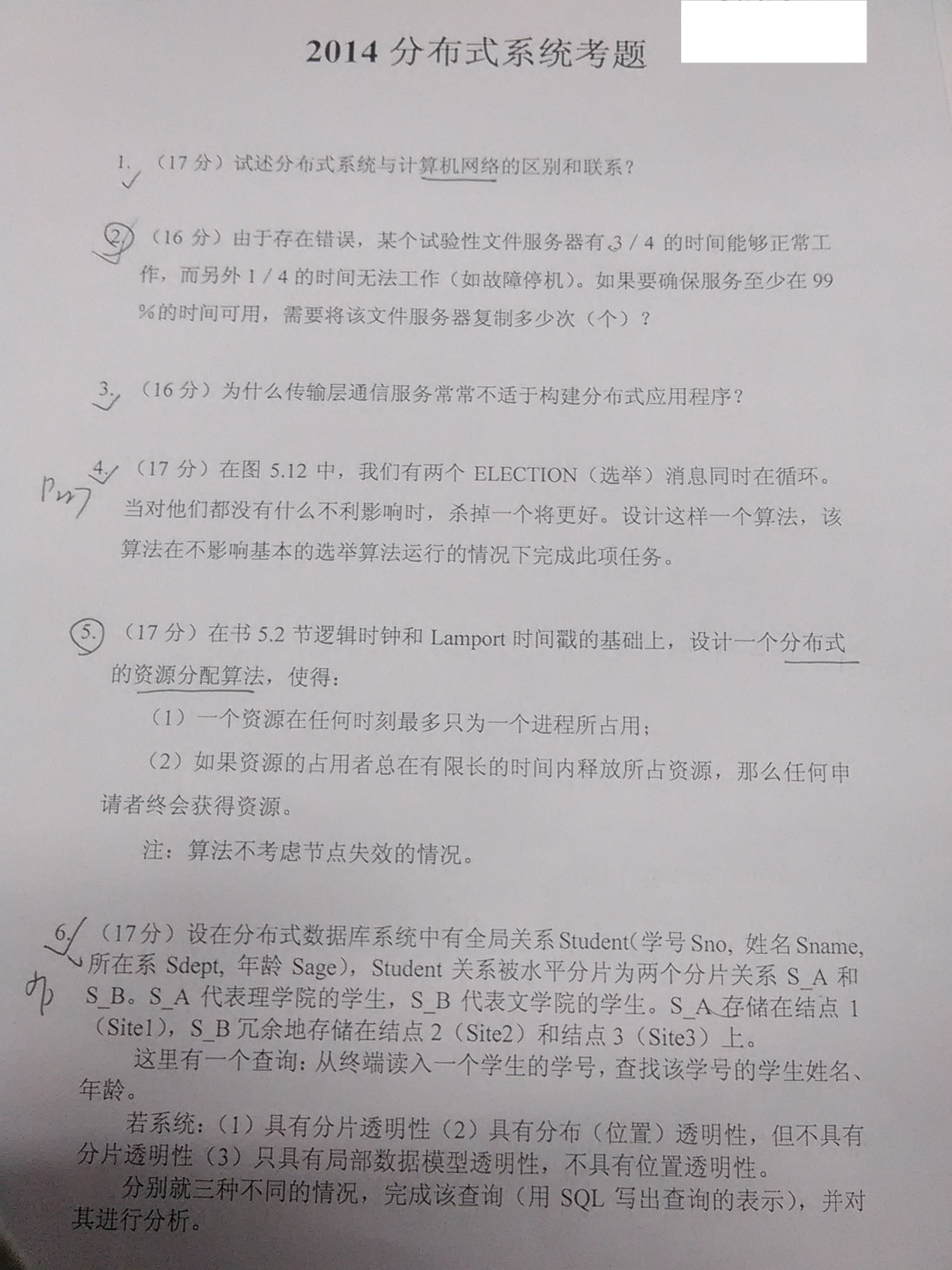
答: 在单线程情况下，命中cache花了15ms，未命中cache花了90ms。加权平均值为2/3 X 15 + 1/3 X 90。这意味着请求花了40ms，服务器没秒可以完成25次。对于多线程，所有磁盘等待都是交迭的，所以每个请求花了15ms， 服务器每秒可以处理66 个请求。

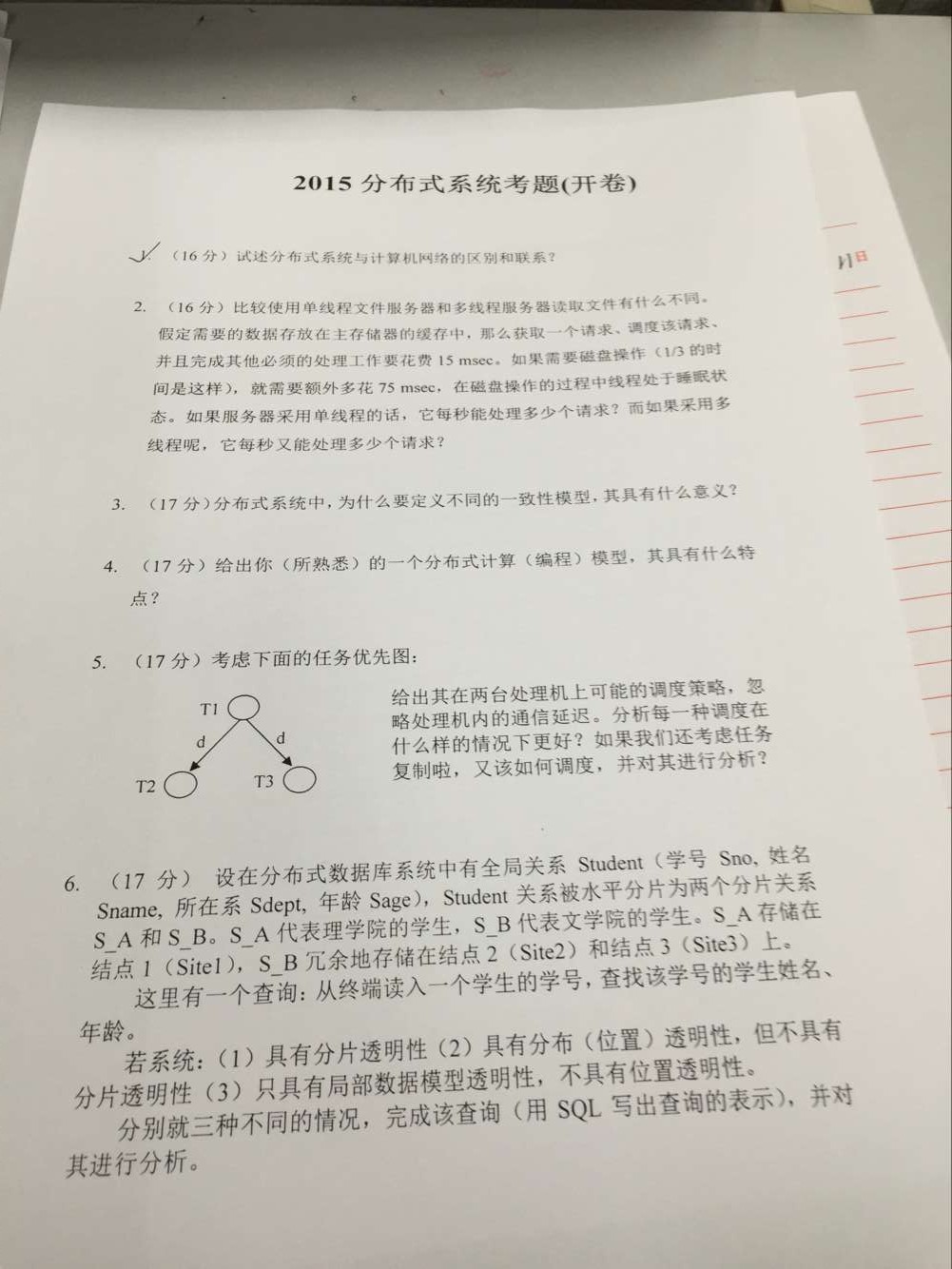
三 没找到

四，









2014年第五题：

(Lamport面包店算法是解决多个线程并发访问一个共享的单用户资源的互斥问题的算法。 由Leslie Lamport（英语：Leslie Lamport）发明。

Lamport把这个并发控制算法可以非常直观地类比为顾客去面包店采购。面包店只能接待一位顾客的采购。已知有n位顾客要进入面包店采购，安排他们按照次序在前台登记一个签到号码。该签到号码逐次加1。根据签到号码的由小到大的顺序依次入店购货。完成购买的顾客在前台把其签到号码归0. 如果完成购买的顾客要再次进店购买，就必须重新排队。

这个类比中的顾客就相当于线程，而入店购货就是进入临界区独占访问该共享资源。由于计算机实现的特点，存在两个线程获得相同的签到号码的情况，这是因为两个线程几乎同时申请排队的签到号码，读取已经发出去的签到号码情况，这两个线程读到的数据是完全一样的，然后各自在读到的数据上找到最大值，再加1作为自己的排队签到号码。为此，该算法规定如果两个线程的排队签到号码相等，则线程id号较小的具有优先权。

把该算法原理与分布式系统相结合，即可实现分步锁。)

一）算法说明

1、若进程（节点）未失效，报文一定能无误地接收到，若失效，则不发信，也不收信。

2、报文传递满足“先发先到”的原则。

（二）报文格式

报文类型 保留 进程号 时间戳 保留

其中：保留部分用作说明申请资源（互斥资源）的类型。

（三）具体规则

1、申请资源的进程s向其它各进程发申请资源的报文，附有时间戳；

2、其它各进程r，收到申诸报文后，若：

(1)不占资源，也不申请资源者，立即回收（报文类型reply0）；

(2)也申请资源，则比较进程了s与自己的时间戳，若T．<T。则回信(replyl)否则回信(reply2)； (3)占有资源的进程，则在使用完资源，才对申请报文的进程补发回信(reply3)．同时释放资源。若其没有收到申请报文，使用完资源也不释放资源。

3、申请资源的进程收到其它所有进程的回信后可以得到：

1）释放资源的进程的报文(reply3) 数目n free；

2）时间戳小于自己时间戳的报文(reply2)的数目n less。

若n—free>n—less，说明系统中的释放出的资源数目多于时间戳比自己小的进程数目，显然可获得资源，否则不能。

因为，不占资源也不申请资源的进程和申请资源且时间戳大的进程，对于申请资源且时间戳小的进程而言，效果是一样的（都不提供资源且不会先得资源），所以可将它们看成一样进行处理即replyO=replyl。

2015年第三题：

一致性模型实质上是进程和数据存储之间的一个约定。即，如果进程同意遵守某些规则，那么数据存储将正常运行。

正常情况下，一个进程在一个数据项上执行读操作时， 它期待该操作返回的是该数据在其最后一次写操作之后的结果。

在没有全局时钟的情况下，精确地定义哪次写操作是最后一次写操作是十分困难的。

作为替代的方法，我们需要提供其他的定义，因此产生了一系列的一致性模型。每种模型都有效地限制了在一个数据项上执行一次读操作所应返回的值。