分布计算环境思考题

一． 通过生成进程来构建并发服务器与使用多线程来构建并发服务器相比有优点也有缺点，请分析这两种方式的优缺点。你认为基于 CORBA 实现的并发服务器是基于生成进程的方法，还是基于多线程的方法？为什么？

第一问参考答题思路：

一般需要答出以下几点：

并发服务器需要同时处理多个请求。接收请求的进程或线程，自己不处理请求，而是将请求传递给某个独立的线程或者其他进程来处理，自身立即返回并等待下一个输入的请求。

采用多进程：优点：1）处理各个请求的进程之间隔离性好；

缺点：1）创建/撤销/切换 处理各个请求的进程的代价大。2）分发器（主进程……）将请求发送到另一个进程的代价大（如果能说明为什么代价大更好）。3）如果各个子进程间需要通信，代价大。

采用线程：优点：1）创建/撤销/切换 处理各个请求的线程的代价小。2）分发器（主线程……）将请求发送到另一个线程的代价小（如果能够说明为什么代价小更好）。3）如果各个线程间需要通信，代价小。

缺点：1）一个线程出问题，可能会影响其他线程。

第一问存在的主要问题是：针对性不够，只是简单罗列进程和线程的优缺点。

第二问参考答题思路：

CORBA：使用多线程技术实现并发服务器。

因为如果采用多进程实现，有以下问题：1）主进程、子进程都需要 ORB 的 Runtime，进程启动/撤销的代价更大； 2）对于远程方法调用来说，请求的参数比较复杂，主进程将请求通过远程方法调用的方式再发送给子进程，开销比较大。3）如果各个子进程间需要通信，远程方法调用代价大，进程间通信代价大。所以如果采用多进程的话实现并发 CORBA 服务器性价比不高。

第二问的主要问题是：理由不充分，如 CORBA 支持远程调用，客户和服务器不在同一

个位置，所以多进程；服务对象由不同语言编写，不能在单一进程中；为支持高并发及高可用，所以多线程或多进程。

二． 为什么传输层通信服务往往不适合用于直接构建比较复杂的分布式应用？目前的解决办法是什么？为什么这样做？

参考答题思路：

首先，说明传输层通信服务提供什么样的能力？只是为端到端连接提供传输服务。

其次分析构建比较复杂分布式应用需要什么样的支持？不仅仅是端到端的通信支持，而且要求具有一些分布透明性，如位置的透明性、访问透明性等，显然，仅仅基于传输层服务，位置、访问透明性等的支持，例如远程对象访问方法的打包拆包等等，都需要应用程序开发者来负责实现，大大加大了应用开发的难度。

目前使用分布计算环境（中间件）来支持相应的分布式应用系统的实现。例如使用CORBA、EJB 支持面向对象的分布式系统的实现。使用消息中间件来支持面向消息的分布式系统的实现。使用 Web Service 来实现 Web 环境下分布式系统的实现。等等（举 2 个或以上例子就好）

这些分布式计算环境解决了相应的分布式应用系统要解决的共性问题，如支持访问、位置透明性，使得分布式应用系统可以更加方便地构建。

主要问题：

(一) 为什么不行，说得太简单，就说了没有支持分布透明性，需要开发人员注意通信的实现，从而导致解决方案的可扩展性很差。

(二) 解决方法单一：RPC、MPI、HTTP、消息队列……

(三) 使用 C/S 模式

(四) 流、各种应用级协议都提到了，就是不提分布计算环境。

(五) 局限在通信一点上

三． DNS 中的高层命名服务器（那些在 DNS 命名空间中接近根的）一般不支持递归式名字解析，为什么？你认为 CORBA 的命名服务使用的是哪种解析方法，为什么？

参考答题思路：

（1） 采用递归方式，对性能影响较大：维持缓存、服务器要等待等等。而基于 DNS 的工作机制，高层服务器要处理的请求量大，对性能要求高。所以……

（2） CORBA 命名服务通过 resolve 方法，根据指定的对象名，返回给相应的对象引用，对于客户来说，这是一次请求得到最终结果的方式，因此可以认为是递归方式。采用这种一次性获得结果的方式，使得客户端编程简单便捷。（也可以说，基于 CORBA 的系统，其名字数量远远小于 DNS 中的量级，即使使用递归，性能上也可以接受）

主要问题：

(一) 说明了 DNS 的工作机制，指明根域名服务器不支持递归，但没有说明为什么。

(二) 没有说明递归方式为什么对性能影响大。

(三) CORBA 的命名服务提供了 Iterator 迭代接口，但这不能说明是迭代解析。

四． 你认为提供百度或谷歌等类似搜索服务的服务器是有状态服务器还是无状态服务器？为什么？你认为搜索服务是有状态服务还是无状态服务？为什么？

参考思路：

我认为….是无状态服务器。因为：同一时刻请求量大，会话状态数据持久化代价过大；使用服务器集群，难以保证客户在类似“下一页”时能够访问相同的服务器，有状态服务器需要同步状态，开销很大。（可以进一步说说采用无状态服务器的优势）而采用无状态服务器，不需要存储会话状态数据，服务器集群可以很方便地进行水平扩展，请求不依赖服务端的信息，任何多次请求不需要必须访问到同一台服务器，服务器副本之间不需要进行状态同步。

我认为搜索服务是有状态服务，因为对于客户来说，使用搜索服务时可以一页一页地显示某次搜索结果，就好像服务器记录了这次搜索会话的总的搜索结果和客户的访问状态一样，能够准确地响应用户要求的某页搜索数据。

或我认为搜索服务是无状态服务，因为……(给出比较能让人信服或者自圆其说的理由)

（注意有状态和无状态服务（器），主要关注的是会话状态。）

问题：

1. 是无状态服务器，因为引入了 Cookie；因为 HTTP 是无状态的

2. 是有状态服务器，因为通常要记录用户的搜索历史，从而定制化地……
3. 是无状态服务，因为 HTTP 请求和以前的没有联系；只需要提供请求参数，不需要提供连接信息
4. 是有状态服务，因为有状态服务可以有效地收集用户的行为信息；可以根据用户的身份、偏好、搜索历史等对搜索结果进行不同的优化、排序等。（可以得分）

五． 组合通信持久性和通信同步性，谈谈微信支持哪几种组合方式。

参考思路：

从通信持久性方面可将通信分为持久通信和瞬时通信，从通信同步性方面可将通信分为同步通信和异步通信。

微信支持持久异步通信，例如在微信群中发送一个大文件时，即使这个文件没有发完，也可以接着在该群中发送其它消息，不用等待这个大文件发完，即消息发送是异步的。进一步地，这个大文件发送时，即使群中有用户不在线，当他们上线时，也可以在群消息中收到这个文件，即消息会存储在微信系统中，是持久化的。因此微信支持持久异步通信。

微信也支持瞬时同步通信。例如使用微信进行语音或者视频通话，要求通信双方必须同时在线，即瞬时通信。该视频或语音通话在发起时，需要对方响应才能继续，是同步的，因此微信支持瞬时同步通信。

问题：

1、 音视频通信到底是同步还是异步通信？
2、 持久异步通信，对持久的理解比较到位，但是对于异步的理解不够到位，比如发送完一条消息后，即使对方不在线，也可继续发送，不用阻塞，因此是异步的；发送方在发送完消息后便可以与其他人通信；
3、 瞬时异步消息，举例，黑名单，A 被拉入黑名单，A 发送消息立刻提示出错，这个例子无法说明是瞬时通信
4、 基于发送的持久同步通信，举例，A 发送消息到 B，几分钟后 B 上线，A 与 B 的时间戳一致，这个例子难以说明是同步通信
5、 不以微信自身举例，只给出各种组合方式的含义，甚至实现原理。