Review of Distributed Computing Technology

2017

**一.** 复习大纲

* 第一章：掌握分布式计算、集中式计算、JAVA 三种类型程序、JAVA 线程等基本概念；理解分布式计算与集中式计算的区别和它们的优缺点等；了解分布式计算与相关计算形式的区别；理解 CAP 理论和了解它在 Nosql 数据库的应用；了解 IPC 的概念、类型、接口原型（API），理解 IPC 的事件同步、死锁及其解决方法；掌握事件状态图和顺序状态图的使用方法。
* 第二章：掌握各种分布式计算范型的概念和原理，包括消息传递、客户-服务器范型、p2p 范型、消息系统、远程过程调用、分布式对象、网络服务、移动代理、云服务模型等，特别理解它们的区别。
* 第三章：理解数据包 SOCKET 和流式 SOCKET 的基本原理、及特别关注它们的区别和适用场景等；掌握两种 SOCKET 的程序设计方法和步骤等；理解客户服务器范型的概念和原理，掌握三层软件架构的开发方法，理解面向连接服务器与无连接服务器的区别、并发服务器与迭代服务器的区别、有状态服务器与无有状态服务器的区别；掌握使用三层软件架构开发 Echo 和

Daytime 服务器程序和客户端的方法（包括 UML 类图设计）。

* 第四章：理解分布式对象范型的体系结构、RMI 的体系结构；掌握一般 RMI 应用、RMI 客户回调应用、桩下载 RMI 应用的开发步骤和方法；了解 RMI 的 3 个高级特征：client callback, stub downloading and security manager。
* 第五章：理解 HTTP 协议的原理、HTTP 请求、HTTP 响应的格式；理解 CGI 的原理（结合图）和作用，掌握通过 FORM 请求 CGI 的两种方法（包括 GET 和 POST）及客户端传递参数给 CGI 的方式；掌握会话状态数据维护的技术

（特别是 cookie 和 session 对象），了解 web 会话的概念；掌握 Applet, Servlet 的基本概念和原理；掌握 Applet 和 Servlet 程序设计的方法；理解 Servlet 与Webservice 的区别；理解 rmi 和 webservice 的区别；理解一般 CGI 和 Servlet 的区别；了解 SSH 框架与应用开发；了解常用的 Web 开发技术。

* 第六章：理解 P2P 范型；理解 P2P 网络分类；了解 P2P 典型应用系统（如

DHT）；熟悉 P2P 编程实践的方法。

* 第七章：了解 Web Services 的体系结构，理解 Web Services 的技术原理和应用场景，了解 SOAP Web Services 的协议栈，理解 SOAP 协议的基本原理， 了解 SOAP Web Services 的开发过程。
* 第八章：了解云计算的概念（三种类型的云服务）；理解云计算的关键技术；熟悉 Google 云计算原理和关键技术；了解亚马逊云服务。
* 第十章：熟悉网络存储、RAID 的基本概念和原理；熟悉云存储技术的分类

（从分布式存储方面看）和应用场景；了解 NoSQL 数据库分类及主流产品； 了解对象存储技术原理；了解存储技术趋势。

* 第十一章：了解大数据概念和特征；熟悉大数据存储平台和区别(Hadoop、

Cassandra、Redis、MongoDB)，重点熟悉 Hadoop 的功能与组件；理解大数据计算模式，重点是 Mapreduce 和 Spark；了解大数据分析处理平台（Impala、HadoopDB）；了解云存储编程和大数据并行计算编程的基本方法。

# 二. 各章知识点总结

第一章：

1）并行计算与分布式计算区别？

所谓分布式计算是一门计算机科学，它研究如何把一个需要非常巨大的计算能力才能解决的问题分成许多小的部分，然后把这些部分分配给许多计算机进行处理，最后把这些计算结果综合起来得到最终的结果。

1. 分布式计算的核心技术是？

进程间通信（IPC）

3）解决进程间通信死锁的两种方法？

超时和多线程（子线程）

1. 分布式系统的 CAP 理论是什么？

一致性，可用性，分区容忍性

第二章：

1. 网络应用中使用的最多的分布式计算范型是？

客户-服务器范型（简称CS范型）

1. 消息传递范型与消息中间件范型异同？

消息传递：一个进程发送代表请求的消息，该消息被传送到接受者；接受者处理该请求， 并发送一条应答消息。随后，该应答可能触发下一个请求，并导致下一个应答消息。如此不断反复传递消息，实现两个进程间的数据交换.

基于该范型的开发工具有Socket应用程序接口（Socket API）和信息传递接口（Message Passing Interface，MPI）等 

消息系统模型（消息中间件范型）可以划分为两种子类型：点对点消息模型（Point-to-point message model）和发布订阅消息模型（Public/Subscribe message model）。

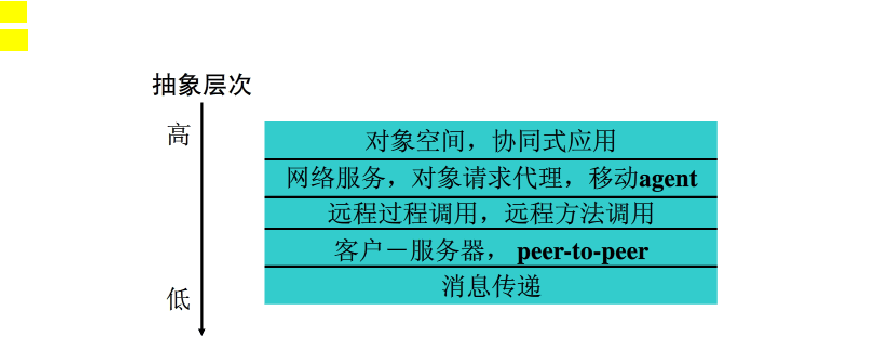
点对点消息模型中，消息系统将来自发送者的一条消息转发到接收者的消息队列中。与基本的消息传递模型不同的是，这种中间件模型提供了消息暂存的功能，从而可以将消息的发送和接受分离。与基本的消息传递模型相比，点对点消息模型为实现异步消息操作提供了额外的一层抽象。如果要在基本的消息传递模型中达到同样的结果，就必须借助于线程或者子进程技术。

发布订阅消息模型中，每条消息都与某一主体或事件相关。对某个事件感兴趣的应用程序可以订阅与该事件相关的消息。当订阅者等待的时间发生时，触发该事件的进程将发布一条消息来宣布该事件或主题。中间件消息系统将这条消息分发给该消息的所有订阅者。发布/订阅消息模型提供了一种用于组播或组通信的强大抽象机制。

1. 一个分布式应用能否使用多个分布式计算范型？

可以

1. 抽象层次最低的分布式计算范型是？



1. Web Services 是基于网络服务范型吗？

是

1. 什么移动代理范型？与其他范型有何不同？

移动代理是一种可移动的程序或对象。在移动代理范型中，一个代理从源主机出发，然后根据其自身携带的执行路线，自动地在网上主机间移动，在每一主机上，代理访问所需要的资源或服务，并执行必要的任务来完成其使命。 移动代理范型为可移动的程序或对象提供了抽象。这种范型不进行消息互换，而是当程序对象在各个参与结点间移动时，携带并传递数据。

1. 分布式对象方向包括哪些子范型？

远程方法调用（RMI）是面向对象版本的RPC（远程过程调用）。在该范型中，进程可以调用对象方法，而该对象可驻留于某远程主机中。与RPC一样，参数可随方法调用传递，也可提供返回值

对象请求代理范型由对象请求者（Object Requestor）、对象提供者（Object）和对象请求代理（ORB）组成。在对象请求代理范型中，进程向对象请求代理发出请求，对象请求代理将请求转发给能提供预期服务的适当对象。

对象请求代理范型与RMI范型非常相似。两者的主要区别在于，对象请求代理范型多了一个对象请求代理，对象请求代理充当中间件角色，作为对象请求者的应用程序可访问多个远程（或本地）对象。对象代理还可以作为异构对象之间的协调者，允许由不同API实现的对象及运行于不同平台上的对象进行交互。

1. 云服务范型有哪三种服务类型？

IaaS 虚拟硬件资源（Amazon）

PaaS 计算、开发环境等平台平台（Google）

SaaS 通过互联网以服务的形式把软件及应用程序按需提供给用户使用的一种范型（SaleForce）

1. ：

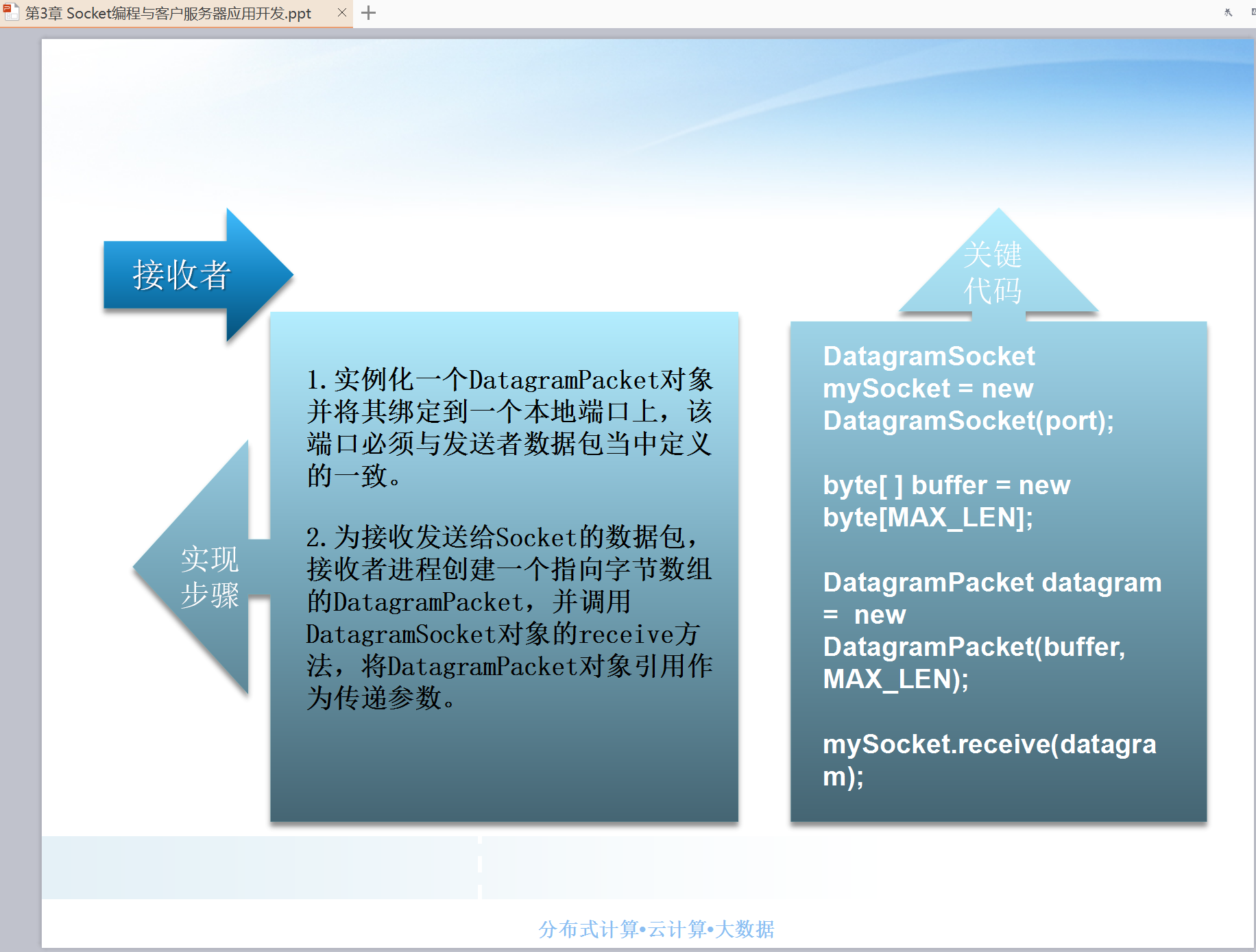
1)数据包 socket 与流式 socket 在传输层有何不同？

数据包socket：UDP，分为 面向连接和无连接

流式socket：TCP，面向连接

1. 基于数据包 socket 的通信过程？





3)基于流式 socket 的通信过程？

P41

1. 数据包 socket 在应用层能否支持面向连接的通信？

是

1. 流式 socket 在应用层能否支持面向连接的通信？

是

1. 并发服务器与迭代服务器的区别？

迭代服务器：不存在重叠的客户会话，在同一时刻，仅能与一位已经接受的客户连接交换数据，非多线程的连接服务器。

并发服务器：能够并行处理多个客户会话。可以使用线程或异步IPC操作来提供并发服务器。常用技术是使用线程，分别使用单个连接socket侦听连接

1. 会话（session ）是什么？

表示服务器和一个客户之间的交互。

服务器管理的服务被多个用户并发访问。

每个客户在使用服务器提供的服务时，都单独与服务器建立一个会话。

在会话期间，客户与服务器进行对话，直到已获取了所需的服务为止.

1. 网络应用(客户服务器应用）开发的三层软件架构是？

软件的三层构架： 

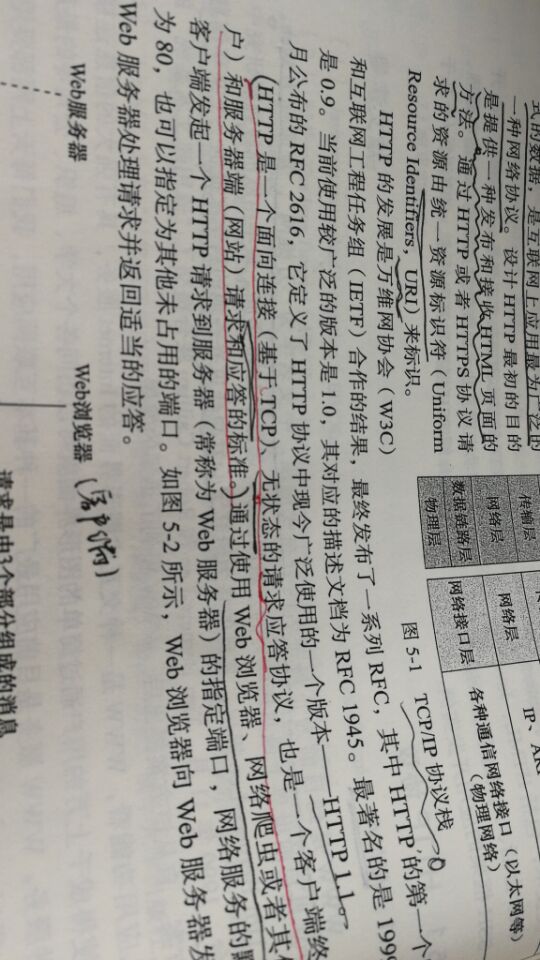
表示层：在服务端，需要用户界面UI来启动服务进程。在客户端需要客户进程提供用户界面，通过该界面客户主机上的用户可请求服务或接受服务器响应。

应用逻辑层：在服务器端，需要根据用户请求进行响应并发送给客户主机。在客户端，需要将用户请求转发给服务器，并将服务器应答显示给用户。 

服务层：支持应用所需的服务，如服务器主机时钟的读出，双方的IPC机制等

1. HTTP 服务器是有状态的还是无状态的?

HTTP协议是面向连接、无状态的



1. 迭代服务器必然是面向连接的服务器吗？

是

第四章：

1）分布式对象范型有哪些具体子范型或协议？

远程方法调用（RMI）是面向对象版本的RPC（远程过程调用）。

在该范型中，进程可以调用对象方法，而该对象可驻留于某远程主机中。与RPC一样，参数可随方法调用传递，也可提供返回值.

对象请求代理范型由对象请求者（Object Requestor）、对象提供者（Object）和对象请求代理（ORB）组成。在对象请求代理范型中，进程向对象请求代理发出请求，对象请求代理将请求转发给能提供预期服务的适当对象。

对象请求代理范型与RMI范型非常相似。两者的主要区别在于，对象请求代理范型多了一个对象请求代理，对象请求代理充当中间件角色，作为对象请求者的应用程序可访问多个远程（或本地）对象。对象代理还可以作为异构对象之间的协调者，允许由不同API实现的对象及运行于不同平台上的对象进行交互。

1. RMI 应用的开发步骤？

RMI应用构建步骤

开发服务器端软件：

1. 指定的远程服务器接口;
2. 实现远程接口;
3. 使用RMI编译器rmic生成stub文件和skelton文件;
4. 创建对象的服务器程序并编译;
5. 激活对象服务器

开发客户端软件：

1. 获取远程接口类文件;
2. 获得存根文件的副本;
3. 开发客户端程序并编译
4. 激活客户端。
5. 基于 RMI 与基于 SOCKET 开发分布式应用有什么区别?
6. socket api的执行与操作系统密切相关，因此执行开销更少。RMI需要额外的软件支持，包括代理和目录服务，这些不可避免地带来运行时开销。
7. RMI api提供了使软件开发任务更为简单的抽象。用高级抽象开发的程序更易理解，因此更易调试
8. 由于运行在底层，SOCKET API通常是平台和语言独立的,RMI则不一定。
9. RMI 对象能否被多个客户端并发访问？

可以

1. RMI 客户回调有什么作用？

回调技术允许对特定时间的发生感兴趣的对象客户在对象服务器上注册自己，以便服务器可以在所等待事件发生时，向对象客户发起一次远程方法调用。 在RMI中，客户回调是允许对象客户在远程回调对象服务器上注册自己，以便服务器可以在所等待事件发生时，向客户发起远程方法调用的一个特征。注意，在客户回调中，远程方法调用编程是双向的（或称全双工的）。

1. 桩下载(Stub downloading)可以解决 RMI 的什么问题？

在分布式对象系统的体系结构中，需要用代理于对象客户的远程方法调用交互。在Java RMI 中，该代理是服务器远程接口stub。 Java RMI提供了客户动态访问stub的机制。使用动态stub下载，客户主机上就不再需要有stub类文件的拷贝了。取而代之的是，stub类可以在客户被激活时，按需从Web服务器传送到客户主机上。

7）桩下载 RMI 应用的执行流程？

查找服务器的RMI注册器，服务器将查询结果返回。

支持stub下载的RMI应用的创建步骤 开发服务器端软件的算法步骤：

1. 为本应用即将生成的所有文件打开一个目录；
2. 在SomeInterface.java中定义远程服务器接口。编译程序，直到没有任何语法错误。
3. 在SomeImpl.java中实现该接口。编译程序，直到没有任何语法错误；
4. 使用RMI编译器rmic处理实现类，生成远程对象的stub文件，可以从当前目录中发现新生成的文件：rmic SomeImpl；
5. 创建对象服务器程序SomeServer.java，编译程序直到没有任何语法错误；
6. 如果希望使用stub下载，将stub类文件拷贝到HTTP主机的相应目录中；
7. 如果使用了RMI注册表，但没有激活，请激活RMI注册表。例如： rmiregistry <端口号，默认1099>

8.也可以在对象服务器程序中使用代码激活RMI注册表；

1. 为应用编辑一个Java安全策略文件，文件名为java.policy，将该文件放在相应目录中，或直接放在当前目录中；
2. 激活服务器，指定希望stub下载的codebase和安全策略文件：

Java –Djava.rmi.server.codebase =http://somehost.someu/edu/stubs/ -Djava.security.polict=java.policy

开发客户端软件的算法步骤：

1. 为本应用即将生成的所有文件打开一个目录；
2. 获取远程方法接口类文件SomeInterface.class的一个拷贝；
3. 开发客户程序SomeClient.java，编译程序生成客户类；
4. 如果不希望stub下载，获取stub类文件的一个拷贝，并将其放在当前目录中；

5.为应用编辑一个Java安全策略文件java.policy，将该文件放在相应的目录或直接放在当前目录中；

1. 激活客户，指定安全策略文件：

java –Djava.security.policy=java.policy SomeClient

1. RMI 客户回调应用开发与一般 RMI 应用开发有什么区别?

与基本RMI体系相比，此时需要两组代理，其中一组代理是服务器远程接口所需的，这和基本RMI体系结构中是相同的；另一组代理用于另外一种接口，即客户远程接口，客户远程接口所提供的方法可以被服务器在回调时调用。

第五章：

1. HTTP 请求和响应命令的组成与格式？

请求格式：

HTTP请求包(GET、POST等请求方法)由三个部分构成，分别是：

方法-URI-协议/版本，

请求头，

请求正文

由一个请求行、若干消息头、以及实体内容3个部分组成，其中的一些消息头和实 体内容都是可选的，消息头和实体内容之间要用空行隔开。

——<command><document address><HTTP version>

——可选头部

——使用post方法的CGI所需的可选数据 

应答格式：

与HTTP请求包相似，由三个部分构成，分别是：

协议-状态代码-描述，应答头，应答正文。

由一个状态行、若干消息头、以及实体内容 3个部分组成，其中的一些消息头和实体内容都是可选的，消息头和实体内容之间要用空行隔开。

响应消息的实体内容就是网页文件的内容，也就是在浏览器中使用查看源文件的方式所看到的内容。

一个使用GET方式的请求消息中不能包含实体内容，只有使用POST、PUT和DELETE方式的请求消息中才可以包含实体内容。

——<protocol><status code><description>

——可能包含几行头信息

——文档本身 响应状态码：

1— 信息类，收到请求

2— 成功类，处理成功

3— 重定向类，需进一步请求

4— 客户端错误

5— 服务器错误

1. HTTP 请求中常用的请求命令有哪些？

GET、POST、HEAD、OPTIONS、DELETE、TRACE、PUT、CONNENT

1. CGI 是什么作用？可以编写 CGI 程序的语言有哪些？

CGI作用：与客户端进行动态的交互，可以通过GET/POST来交互数据，交由CGI处理。 GET方式是直接在地址栏传输，POST在数据文件内部。GET方式方便，但是有长度限制而且数据裸露不安全。

CGI程序能够用Python, PERL, Shell, C or C++等语言来实现。

1. 在 web Form 中请求 CGI 有哪两种方法？它们传递参数给 CGI 的方式分别是？

WEB表单从浏览器请求服务器的方法有GET和POST两种（与HTTP协议中的两种请求方 法相一致）。如果方法（METHOD属性值）是GET，则CGI程序就从环境变量QUERY\_STRING中获取Form数据；若方法是POST，则CGI程序就从标准输入（stdin）中获取Form数据。

1. 在客户端维护会话状态数据和服务器端维护会话状态数据各有什么优缺点？

cookies 和 session

cookies：

优点：极高的扩展性和可用性

1. 通过良好的编程，控制保存在cookie中的session对象的大小。
2. 通过加密和安全传输技术（SSL），减少cookie被破解的可能性。
3. 只在cookie中存放不敏感数据，即使被盗也不会有重大损失。
4. 控制cookie的生命期，使之不会永远有效。偷盗者很可能拿到一个过期的cookie。

缺点：

1. Cookie数量和长度的限制。每个domain最多只能有20条cookie，每个cookie长度不能超过4KB，否则会被截掉。

2. 安全性问题。如果cookie被人拦截了，那人就可以取得所有的session信息。即使加密也与事无补，因为拦截者并不需要知道cookie的意义，他只要原样转发cookie就可以达到目的了。

1. 有些状态不可能保存在客户端。例如，为了防止重复提交表单，我们需要在服务器端保存一个计数器。如果我们把这个计数器保存在客户端，那么它起不到任何作用。

session对象带给我们的一些好处,也就是优点:

  1.session对象可以在页面间传递一个变量,它传递数据比request的querystring方法要方便.

  2.session对象可以让网页记住用户名,实现网页的用户化, 从而赢得用户的喜好.

  3.在asp-vbscript中,你可以直接使用它,不需要去定义,使用完毕后,也不用去管它.

其次: session对象在使用时也存在一些缺点:

  1.一但遇到用户的浏览器设置为不支持cookies的时候,那么session也就没有办法使用.

  2.session是程序执行自动生成的,在这里我们只是存放了单个的用户名,内容很少,如果存放的是大量的数据或者是一个recordset对象返回的记录集,而且很多网站都没有设置session的有效期,都是20分钟,那么机器的负担就跟着一起增加.

  3.由于可以直接使用它,不需要去定义,所以一但使用过多,每个session变量的含义变得不很清晰。

1. JAVA 的三种类型程序的区别？

Application

―Java应用程序”是可以独立运行的Java程序。

由Java解释器控制执行。 

Applet

  ―Java小程序”不能独立运行（嵌入到Web页中）。

由Java兼容浏览器控制执行。 

Serverlets

是Java技术对CGI 编程的解决方案。

是运行于Web server上的、作为来自于Web browser 或其他HTTP client端的请求和在server上的数据库及其他应用程序之间的中间层程序。

 Servlet 与 Applet 的比较 相似之处：

• 它们不是独立的应用程序，没有main()方法。

• 它们不是由用户或程序员调用，而是由另外一个应用程序(容器)调用。 •

它们都有一个生存周期，包含init()和destroy()方法。

1. Servlet 程序与一般 CGI 的区别？

Servlet与CGI的比较

与传统的CGI和许多其他类似CGI的技术相比，Java Servlet具有更高的效率，更容易使用，功能更强大，具有更好的可移植性，更节省投资。在未来的技术发展过程中，Servlet有可能彻底取代CGI, 它有如下特点:

 方便  功能强大  可移植性好  节省投资

1. 最常用的两种 WEB 会话跟踪技术（维护）是什么？

cookie是一小段文本信息，伴随用户请求在Web服务器和浏览器之间传递，Cookie存放于客户端

 Session： Web服务器为客户端开辟的一块存储空间，用于存放状态信息

1. 动态网页开发语言有哪些？

PHP , JSP , ASP

1. SSH 框架的作用是？

SSH框架:用于构建灵活、易于扩展的多层Web应用程序。

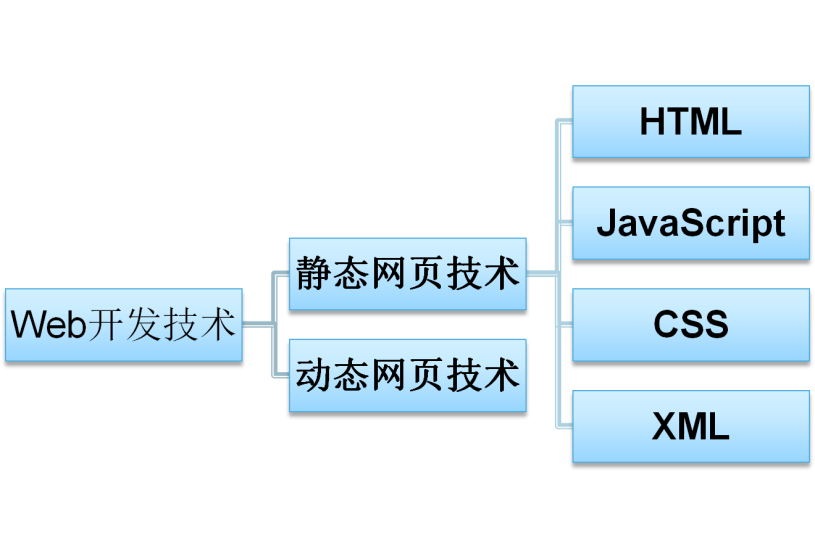
表示层：JSP交互，Struts分派MVC分离

 业务逻辑层：spring IoC容器分派DAO

数据持久层：hibernate（ORM对象关系映射） 处理DAO并返回（数据库操作）

域模块层（实体层）

1. Web 开发技术有哪些？



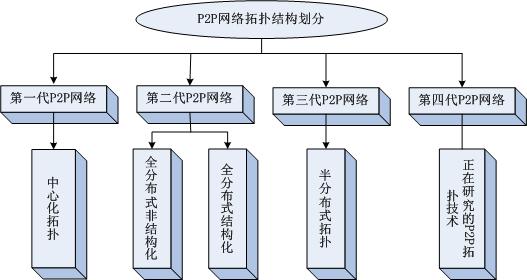
动态：PHP , JSP , ASP

第六章：

1）什么是 P2P 范型？

peer-to-peer范型源于P2P网络（又称为对等计算网络），简称为P2P范型。P2P网络是无中心服务器，依赖用户群交换的互联网体系。 在peer-to-peer范型中，各参与进程的地位是平等的，都具有相同的性能和责任（因此，称它们为peer）。每个参与者（进程）都可以向另一个参与者发起请求和接收响应。在一个基于P2P范型的分布式应用中，每一个参与的进程往往既承担的服务器进程的角色（资源提供者），又承担了客户进程的角色（资源请求者）。

1. P2P 网络的分类有哪些？



 Napster是典型的P2P中心化拓扑结构系统

典型的全分布式非结构化拓扑的P2P网络有Gnutella。

全分布式非结构化拓扑结构的P2P网络是一种重叠网络（原有+虚拟11点对应 

 全分布式结构化拓扑采用分布式散列表（DHT）

典型的DHT网络案例有Tapestry、Pastry、Chord、CAN等。 

 半分布式拓扑选择了性能较高的结点作为超级结点

KaZaA就是一款典型的半分布式拓扑的P2P文件共享软件。

1. 常用的 DHT 网络系统有哪些？

典型的DHT网络案例有Tapestry、Pastry、Chord、CAN等。

第七章：

1. 开发 Web Services 的编程语言有哪些？

 都可以。

 WebService是一种跨编程语言和跨操作系统平台的远程调用技术。通过Web Services可以使运行在不同机器上的不同应用无须借助附加的、专门

的第三方软件或硬件，就可以相互交换数据或进行集成。因此，无论应用之间采用什么语言、平台或内部协议，都可以方便地进行数据的交换。

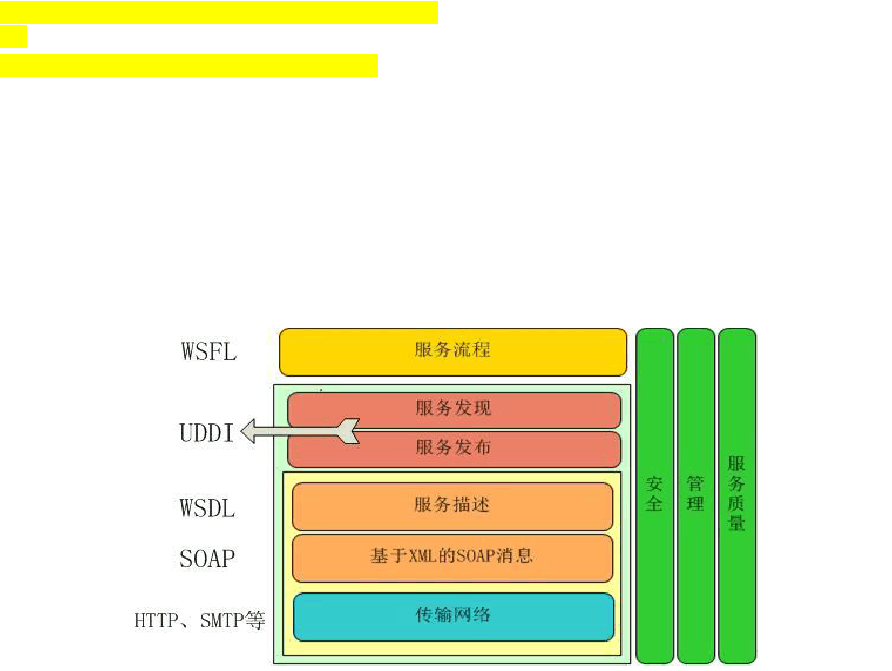
1. Web Services 的请求和响应是基于什么协议传输的？

XML格式封装和基于http协议传输

1. 基于 SOAP 的 Web Services 的相关协议有哪些？

• 基于SOAP的Web Services主要包括SOAP、WSDL、UDDI等技术。

 • 协议栈



1. SOAP 消息是由哪些元素组成的 XML 模式？

SOAP Envelope

SOAP Header

SOAP Body    S

OAP Fault

第八章：

1. 云计算有哪三种服务类型？

基础设施即服务层（IaaS, infrastructure as a service）、

 平台即服务层（PaaS, platform as a service）、

 软件即服务层（SaaS, software as a service）

1. 云计算有哪些关键技术？

云计算关键技术

 体系结构 

数据存储 

计算模型 

资源调度

 虚拟化技术

1. Google 云计算平台有哪些关键技术？

Google云计算平台的几个关键技术

文件存储，Google File System，GFS 

并行数据处理MapReduce 

结构化数据表BigTable 

分布式锁Chubby

1. 亚马逊提供哪些云计算服务？

亚马逊提供的云计算服务

• 弹性计算云EC2

• 简单存储服务S3

• 简单数据库服务Simple DB

 • 简单队列服务SQS

• 弹性MapReduce服务

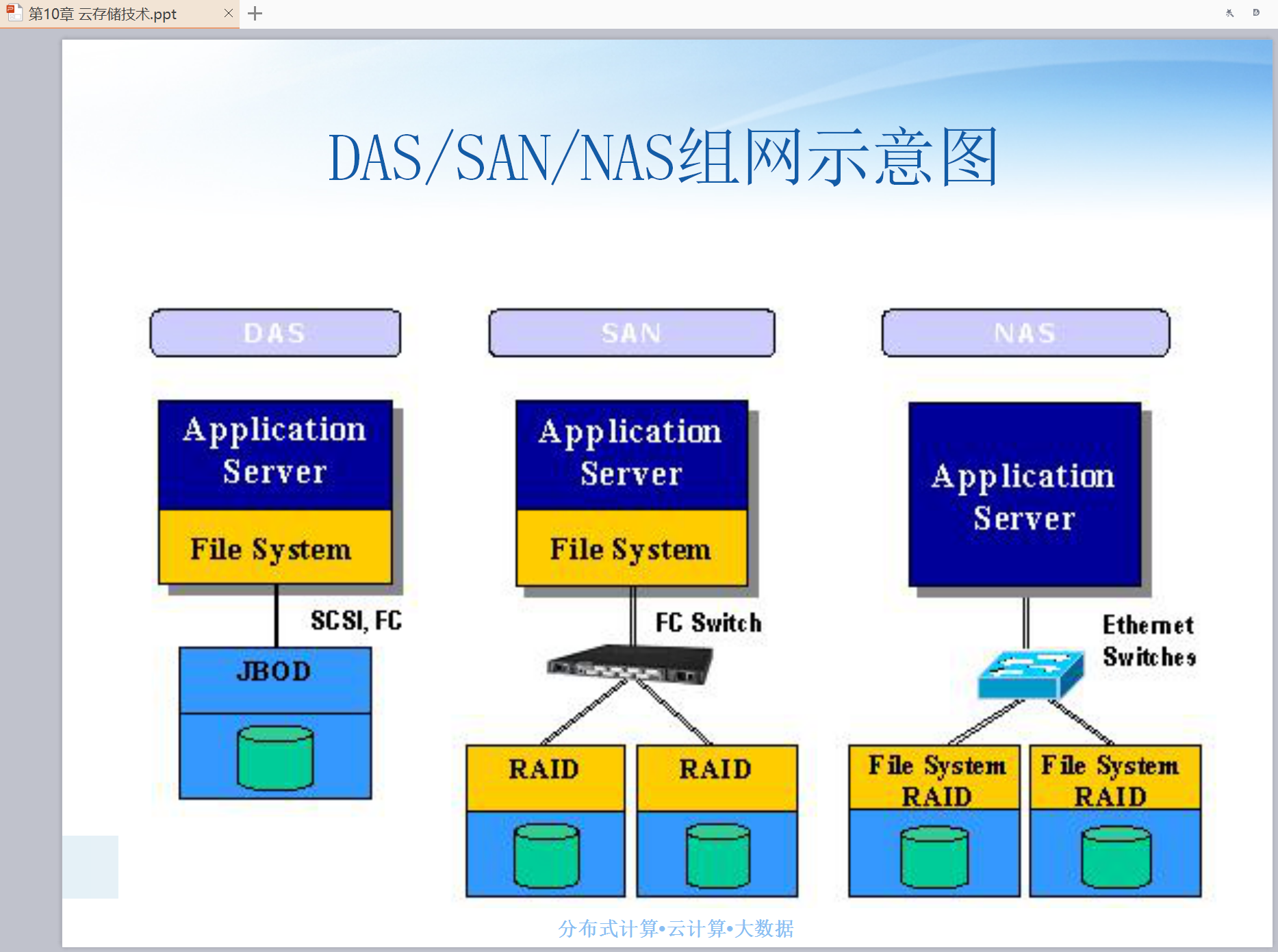
 • 内容推送服务CloudFront

• AWS导入/导出

 • 关系数据库服务

第十章：

1. 网络存储有哪三种组网形态？



1. 磁盘阵列有哪些 RAID 级别？

RAID0：条带 C=(磁盘数量\*容量)

RAID1：镜像 C=(磁盘数量/2)\*容量

RAID3：条带+校验盘   C=(磁盘数量-1)\*容量

RAID5：条带+分布式各1校验（XOR）

RAID6：允许2磁盘损  C=(磁盘数量-2)\*容量

RAID10：1+0          C=(磁盘数量/2)\*容量

RAID50：5+0          C=(磁盘数量-1)\*容量

1. 分布式存储有哪些分类？

目前业界两种主流云存储技术解决方案：基于虚拟化技术和分布式存储。

分布式存储:

分布式块存储

分布式文件存储

分布式对象存储

分布式表存储

1. 对象存储与传统存储有何不同？

在传统的存储系统中用文件或块作为基本的存储单位，块设备要记录每个存储数据块在设备上的位置； 

而在对象存储系统中，对象是数据存储的基本单元，Object维护自己的属性，从而简化了存储系统的管理任务，增加了灵活性，在存储设备中，所有对象都有一个对象标识，通过对象标识OSD命令访问该对象

1. 存储技术有哪些趋势？

数据优化技术将成为热点（精简配置、数据重删、分层存储、数据压缩）

• 存储虚拟化

• 固态硬盘

• 重复数据删除

• SOHO存储

• ROBO存储

• 语义化检索

• 存储智能化

第十一章：

1. 大数据存储平台有哪些？Hadoop 的功能与组件？

HDFS HBase (HADOOP) Cassandra Redis MongoDB

1. 大数据计算模式 Mapreduce 和 Spark 有何异同？

MapReduce是Google公司提出的一种用于大规模数据集（大于1TB）的并行运算的编程模型。 Spark 是一种与 Hadoop 相似的而又强于Hadoop的开源集群计算环境，由加州大学伯克利分校 AMP 实验室开发。Spark 启用了内存分布数据集，除了能够提供交互式查询外，它还可以优化迭代工作负载。

1. 大数据分析处理平台有哪些？

Impala  HadoopDB

1. 大数据研究趋势有哪些？

数据的不确定性与数据质量

跨领域的数据处理方法的可移植性 

数据处理的时效性保证——内存计算 

对于流式数据的实时处理

# 三. 样题

1. **Fill in the blank answers:**
2. Please give five kinds of distributed computing paradigm:（1）客户/服务器范型 (2) 消息传递范型 (3) Peer-to-Peer范型 (4) 网络服务范型 (5) 云服务范型. (消息系统范型、分布式对象范型、移动代理范型)
3. The core technology of distributed computing is (1) 进程间通信（IPC）
4. When communication is from one process to a single other process, the IPC is said to be a（1）

单播（unicast） . When communication is from one process to a group of processes, the IPC is said to be a (2) 组播（multicast）

1. Sockets that use UDP for transport are known as（1） 数据包Socket（Datagram Socket） , while sockets that use TCP are termed (2) 流式Socket　（Stream Socket）
2. A connection-oriented server can be threaded so that it can service multiple clients concurrently. Such a server is said to be a（1）并发服务器( Concurrent server) . An unthreaded connection-oriented server is said to be an (2) 迭代服务器(Iterative server).
3. Java RMI’s three advanced features are: （1） 客户回调（client callback） (2) 桩下载（stub downloading） (3) 安全管理器（security manager）
4. A web form has two methods（: server.

1） PSOT (2) GET sending the query string to the

1. In the Java RMI architecture, the server-side proxy and the client-side proxy are （1）

Skeleton and (2) Stub .

1. （1） 消息传递范型 is the most fundamental paradigm for distributed applications.
2. （1） 客户/服务器范型 is the best known paradigm for network applications.
3. In（1） Peer-to-Peer 范型 paradigm, the participating processes play equal roles.
4. There are two subtypes of message system models: （ 1 ） 点对点消息模型（Point-to-point message model） and (2) 发布订阅消息模型（Public/Subscribe message model） message model.
5. （1） Java RMI is the object-oriented equivalent of RPC.
6. （1） Mobile Agent paradigm offers the abstraction for a transportable program or object.
7. In the stream socket API, the （1） ServerSocket class is for the establishment of connections, while the (2) Socket class is for the transfer of data.
8. In the basic socket API, the send operation is（1）noblocking and the receive operation is (2) blocking . （noblocking or blocking）
9. In the client-server model, we use the term（1） 会话（Session） to refer to the interaction between the server and one client.
10. The functionalities of a distributed application can be classified in three layers: （1）

表示层 (2) 应用层 (3) 服务层

1. In the development RMI application, the stub and skeleton are generated from the implementation of a（1） remote interface using a tool: (2) rmic .
2. **True or False.**

( F ) 1. Stream sockets can support both connectionless and connection-oriented

communication at the application layer.

( T ) 2. Datagram sockets can support both connectionless and connection-oriented communication at the application layer.

( F )3. Iterative server can service multiple clients concurrently.

( T ) 4. Concurrent server can service multiple clients concurrently.

( T ) 5. The distributed objects paradigm is action-oriented.

( T ) 6. The message-passing paradigm is data-oriented.

( F ) 7. Client-server is the most fundamental paradigm for distributed applications.

( T ) 8. HTTP is a connection-oriented, stateless protocol. ( T ) 9. HTTP is a text-based request-response protocol.

( T ) 10. The CGI script can dynamically generate web contents.

( F ) 11. Sending the query string to the CGI program with the GET method, the CGI program will receive the encoded form input from the standard input.

( F ) 12. Sending the query string to the CGI program with the POST method, the CGI program will receive the encoded form input from environment variable.

( T ) 14. Callback allows the server initiates a remote method invocation to the client.

( F ) 15. In the stateless server, the session state information is maintained by the client, so it is more complex to design and implement than the stateful server.

( T ) 16. A CGI program can be written in Perl.

( F ) 17. A CGI program can be written in SQL.

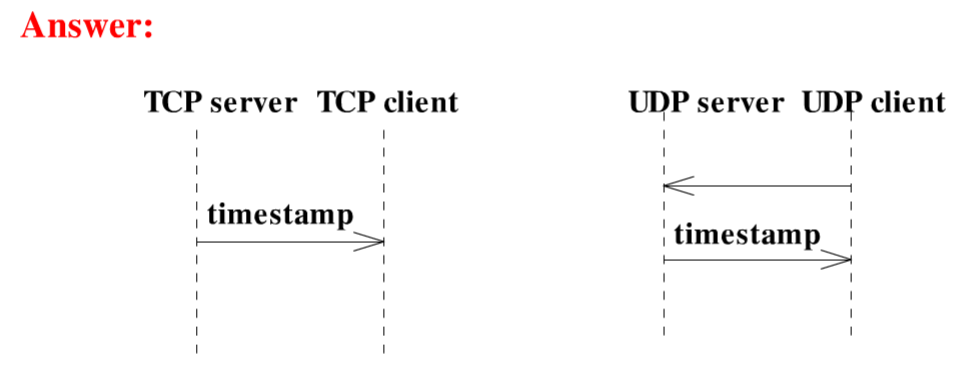
( T ) 19. Well known Internet services include HTTP, FTP, DNS, are client-server applications.

( F ) 20. Well known Internet services include HTTP, FTP, are peer-to-peer applications.

( T ) 21. The Accept() method of ServerSocket class is a blocking operation

( F ) 22. The Accept() method of ServerSocket class is a nonblocking operation

1. **Question**
   1. Draw a sequence diagram for the daytime protocol.



* 1. Is it possible for a daytime client to be blocked indefinitely? Explain.

会。例如:

1. 客户端启动一个阻塞的receive
2. 没有设置该receive的超时时间

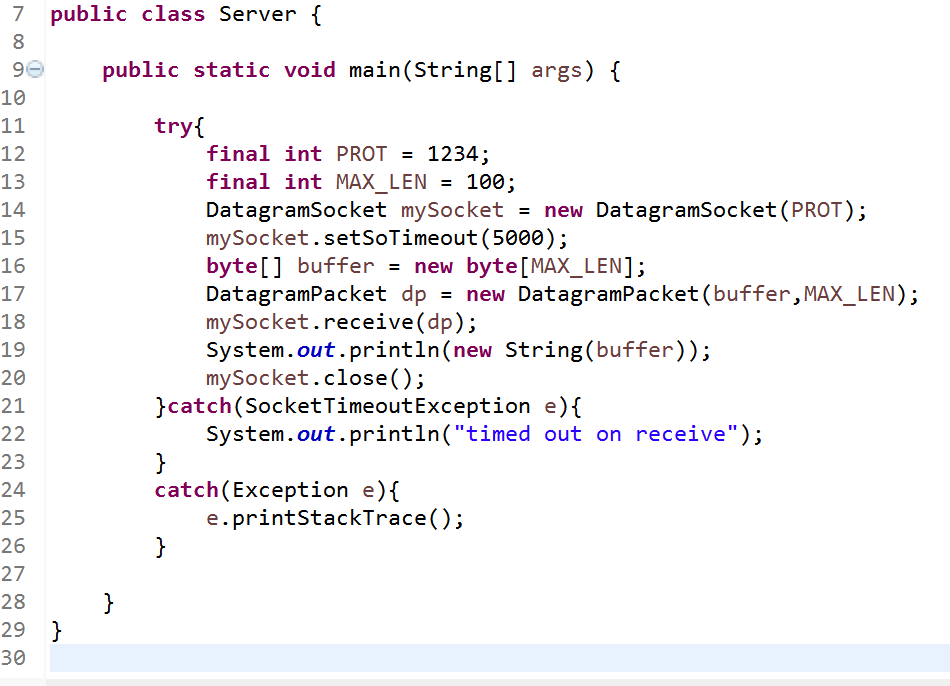
则客户端会无限期阻塞。

* 1. Process 1 sends three messages successively to process 2. What is the possible order in which the messages may arrive at process 2 if
     1. Connectionless socket is used to send each message?
     2. connection-oriented socket is used to send each message?

a： 123、132、213、231、312、321

b： 123

* 1. Write a Java code fragment that may appear in a main method to open a datagram socket for receiving a datagram of up to 100 bytes, timing out in 5seconds. If a timeout does occur, a message “ timed out on receive” should be displayed on screen.



* 1. Using the three-tier software architecture presented in this chapter, design and implement a client-server suite for the following protocol (it is not a well-known service): Each client sends to the server a name. The server accumulates the names received from successive clients (by appending each, with a linefeed („\n‟), to a static string). Upon receiving a name, the server sends the names that it has collected to the client. The figure below illustrates the sequence diagram of a session of the protocol.

**Name server client 1**

**client2**

**client 3**

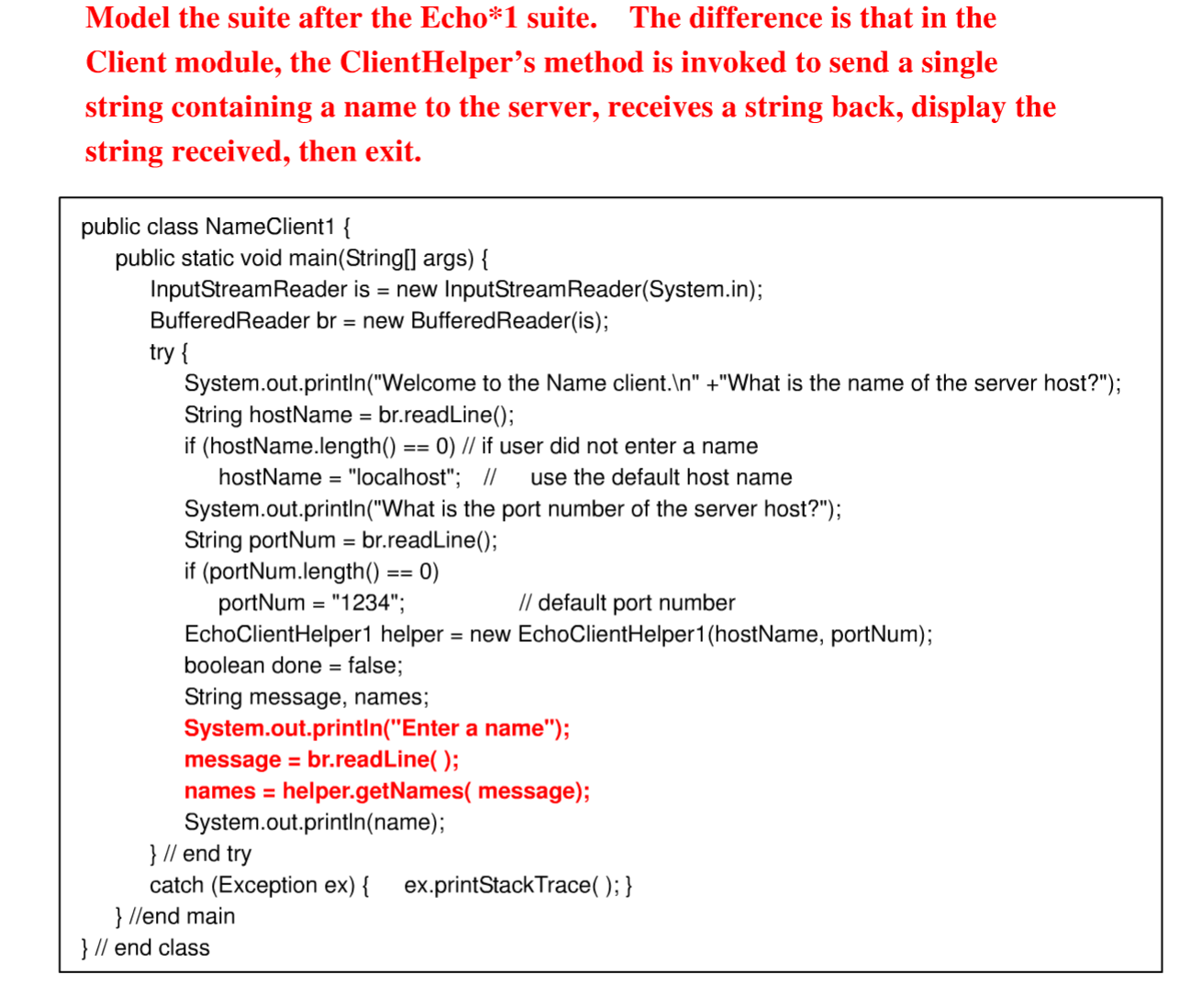
|  |  |  |
| --- | --- | --- |
| **Matt** |  |  |
| **Matt** |
| **Cathy** |
| **Matt\nCathy** |  |
| **John** |  |
| **Matt\nCathy\nJoh** | **n** |  |
|  |  |  |

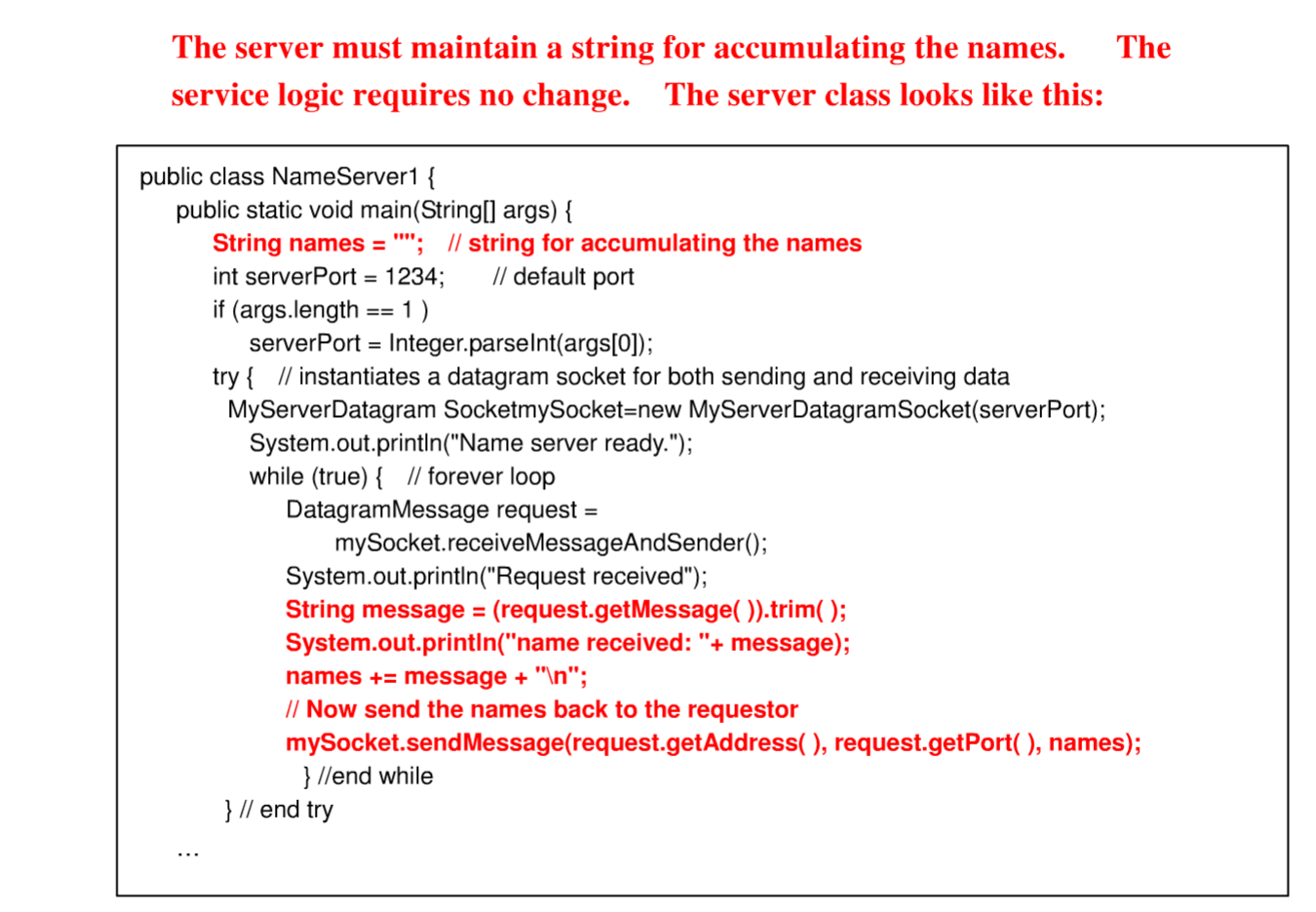
* + 1. Is this server a stateful server? If so, what kind of state information (global or session) does it maintain?

是的，全局信息。

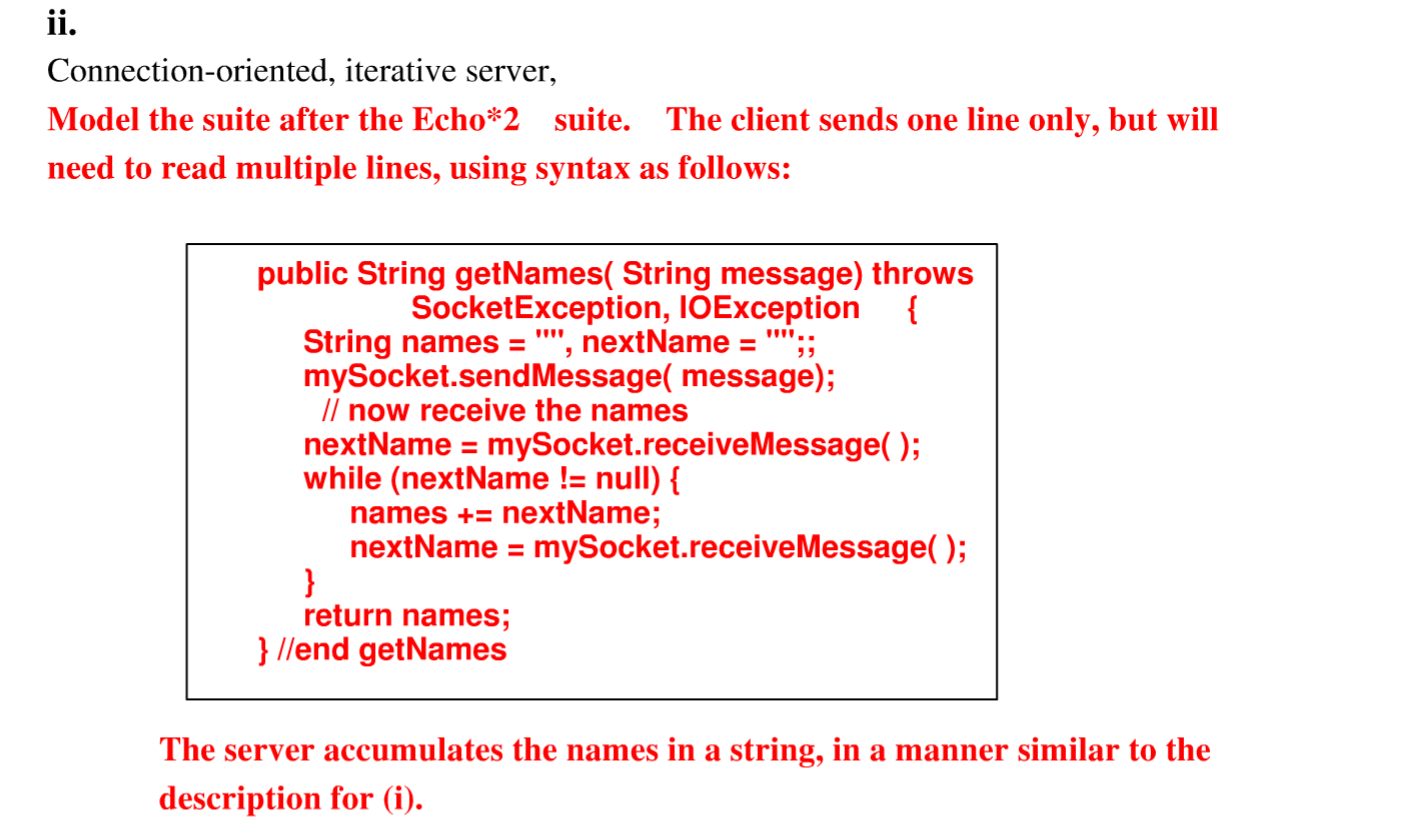
Yes，global。

* + 1. Create one or more of the following suites for the protocol:
       1. Connectionless server;

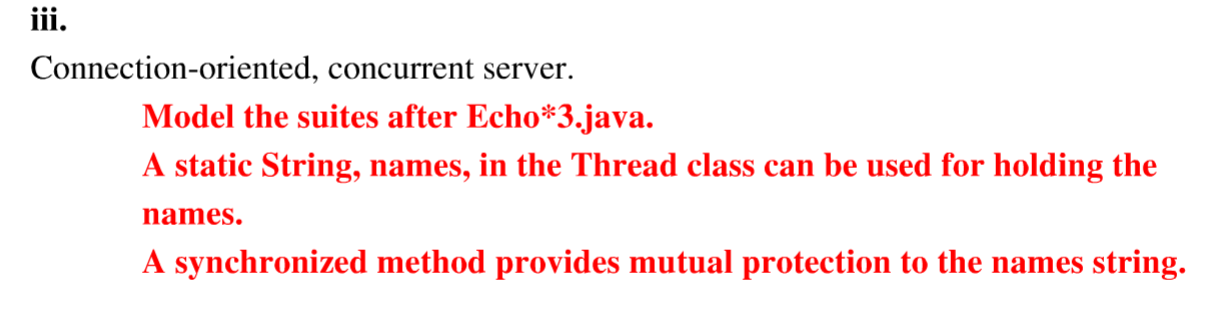


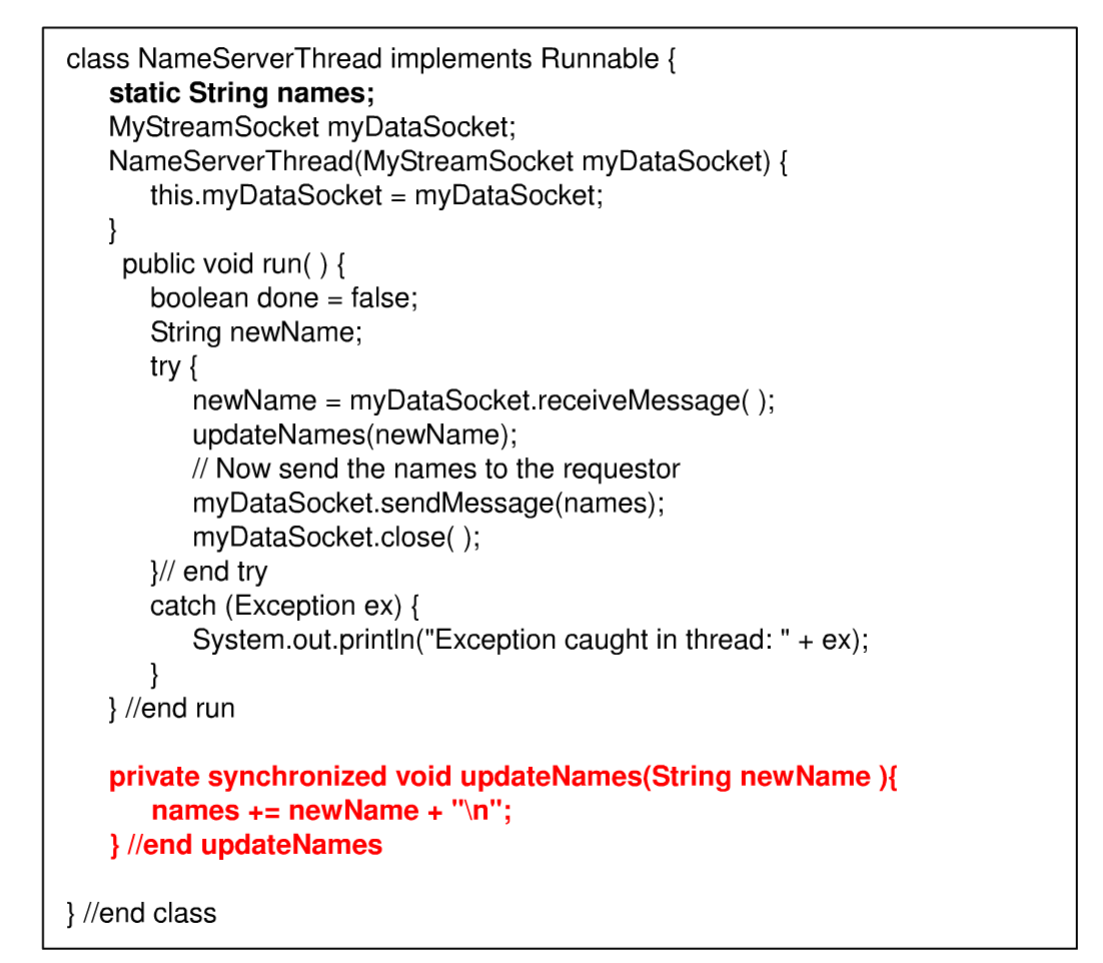


* + - 1. Connection-oriented, iterative server,

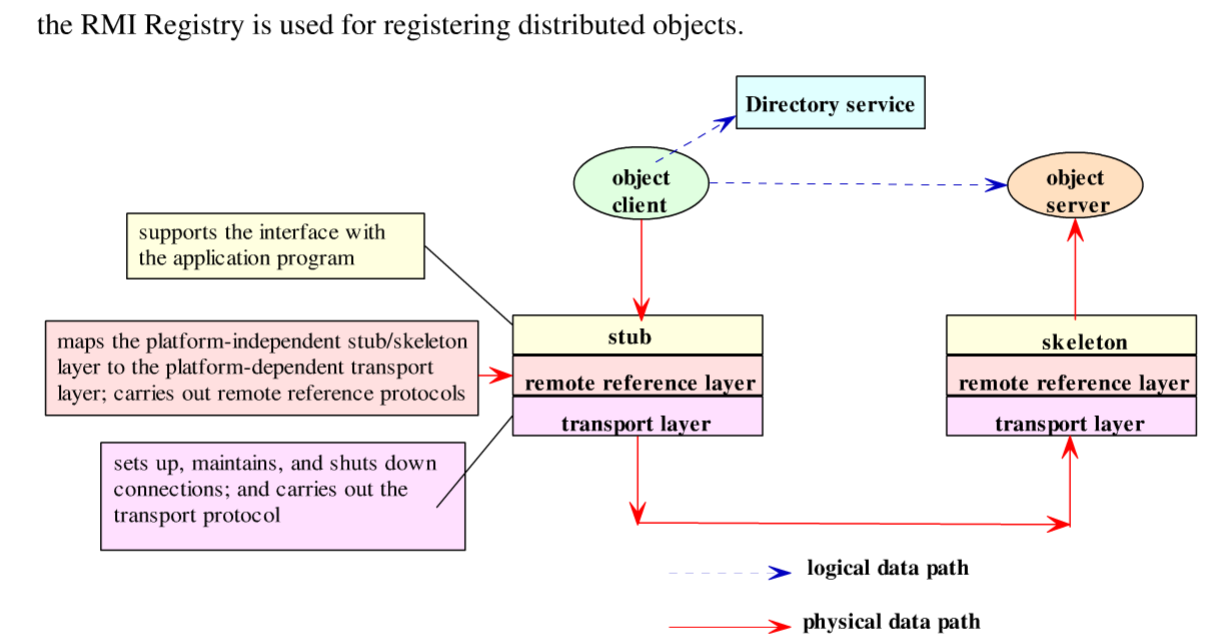


* + - 1. Connection-oriented, concurrent server.

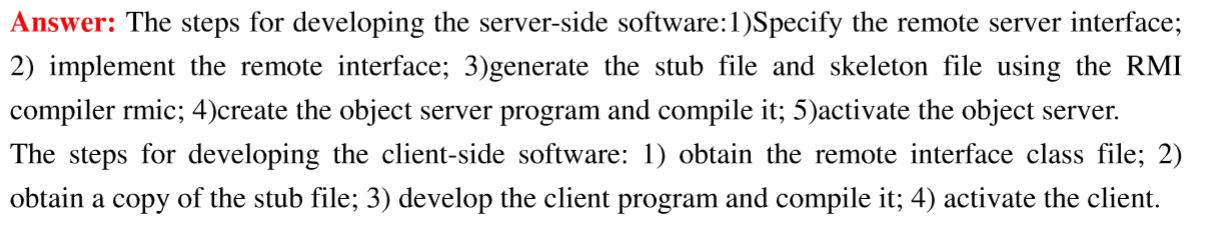




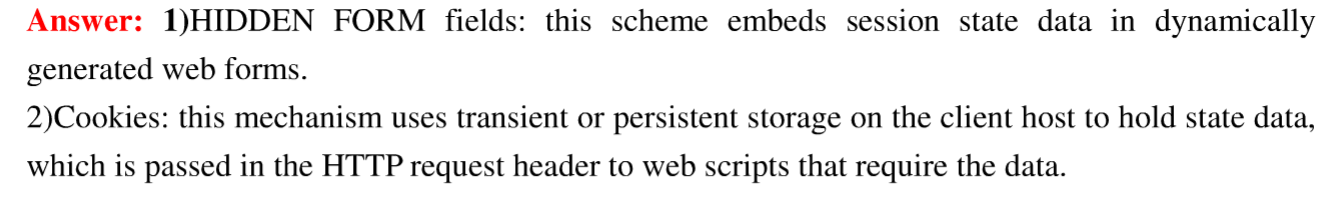
* 1. Please describe the Java RMI architecture. What is the role of the RMI Registry?



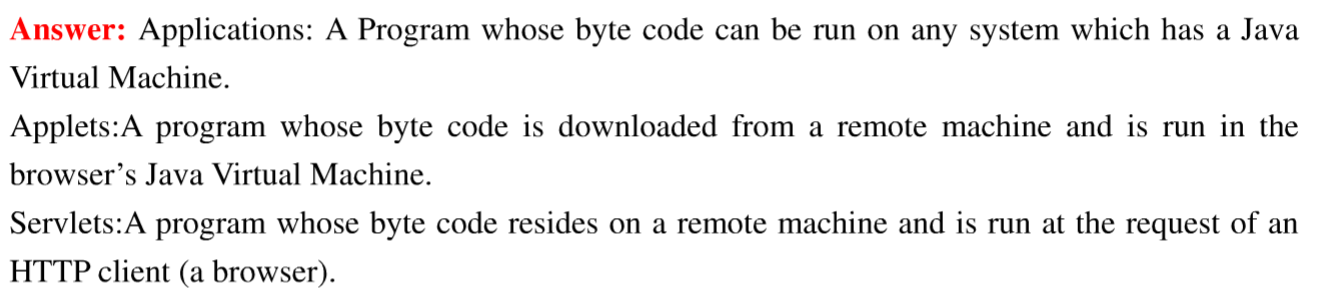
* 1. Please describe the steps for building an RMI application.



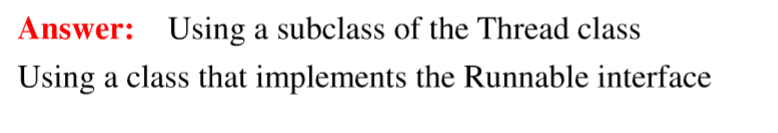
* 1. Please describe two schemes that make use of client-side facilities to maintain session data.



* 1. Please describe three types of Java programs.



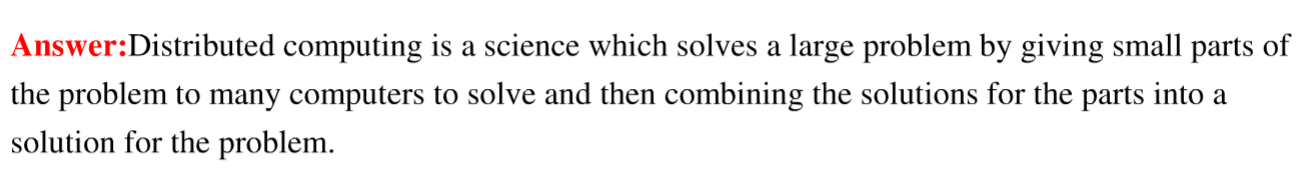
* 1. Please give two ways to create a Java thread.



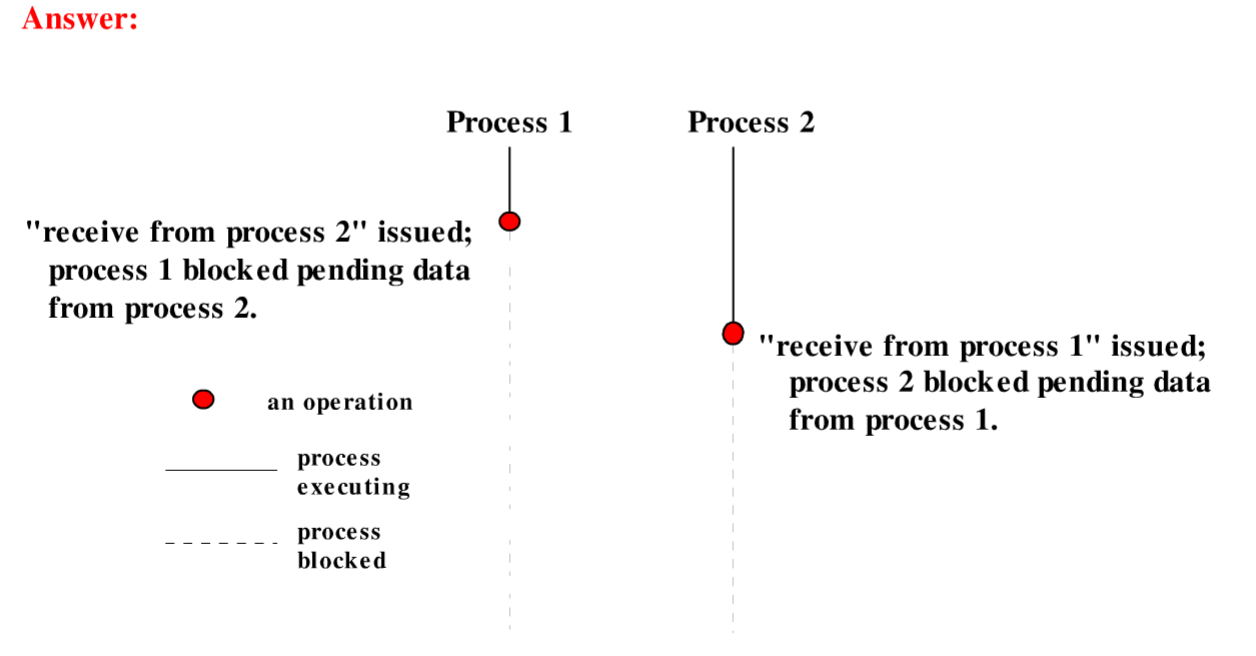
* 1. Please give two examples of distributed computing application.



* 1. Please describe the concept of distributed computing



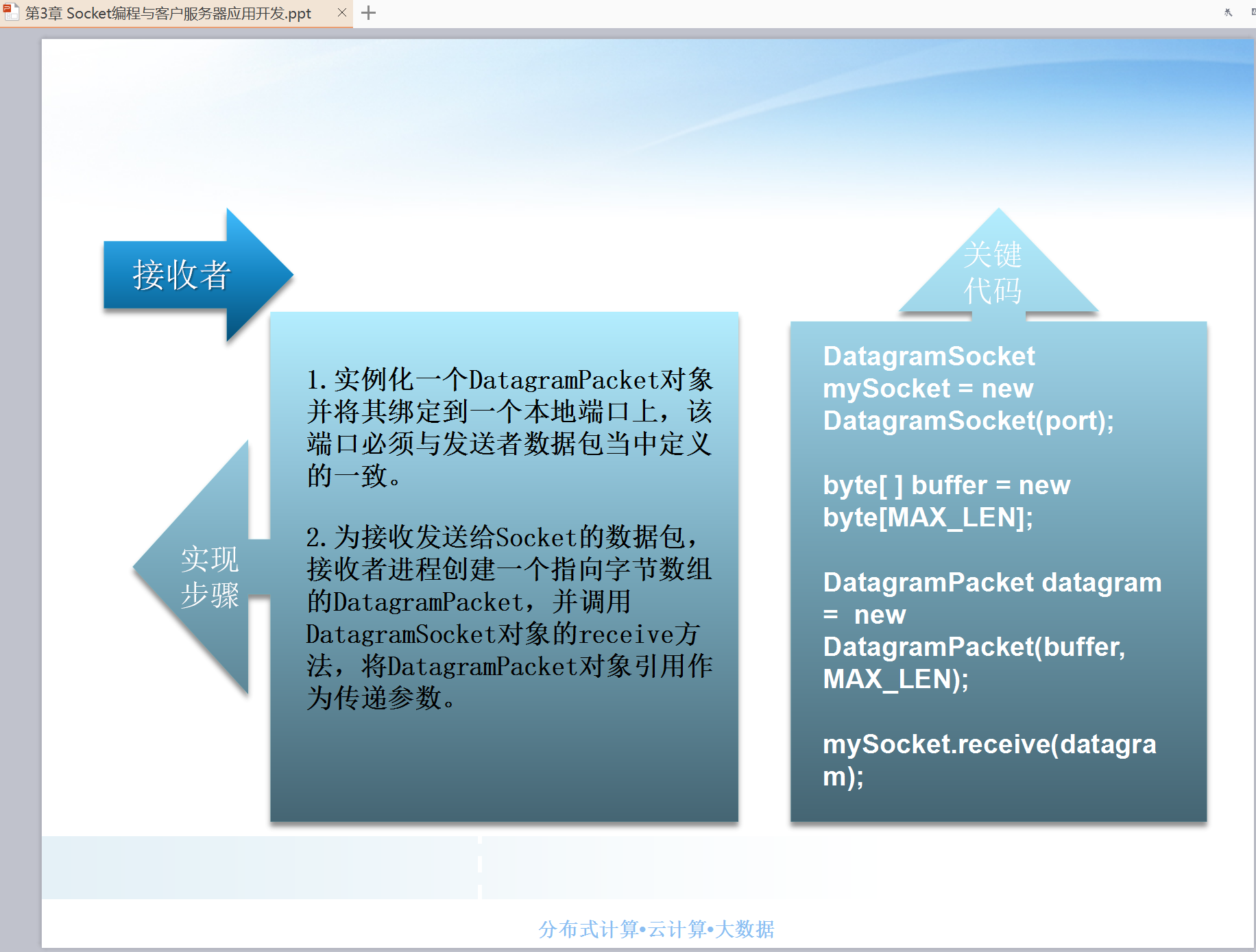
* 1. Please give an example of a deadlock during interprocess communications(involving send/receive operations).



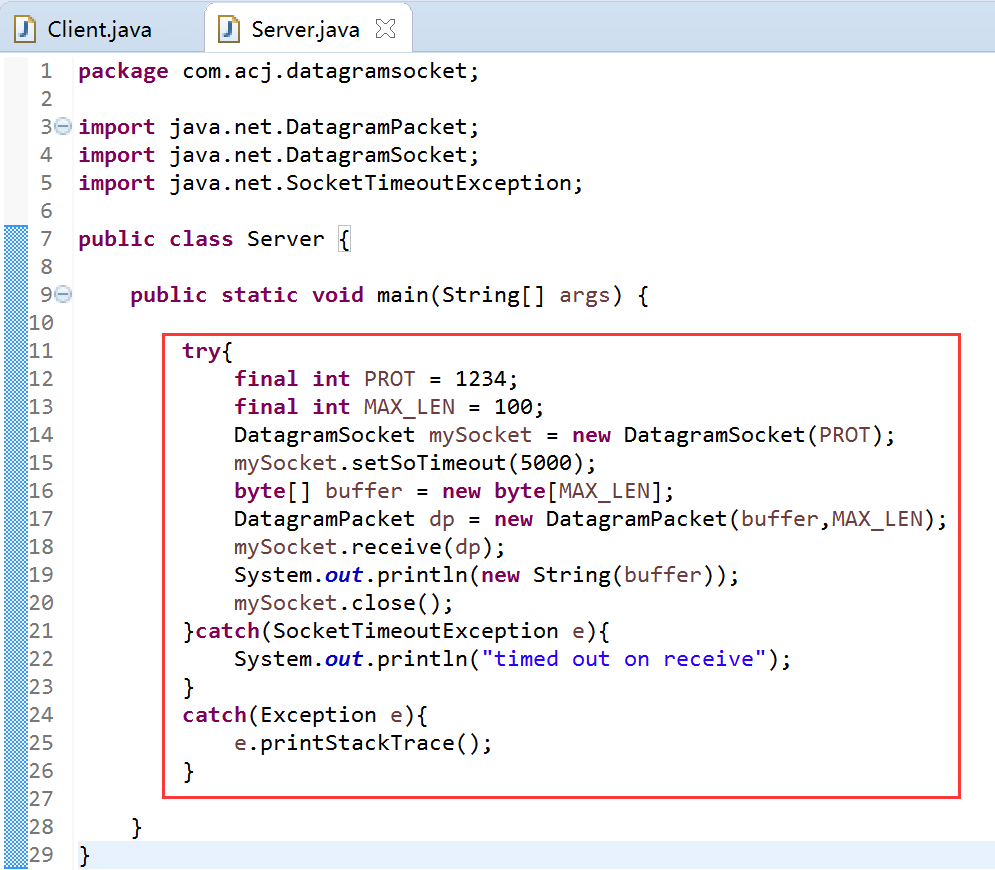
# 四. 编程API

1.数据包Socket API的JAVA实现



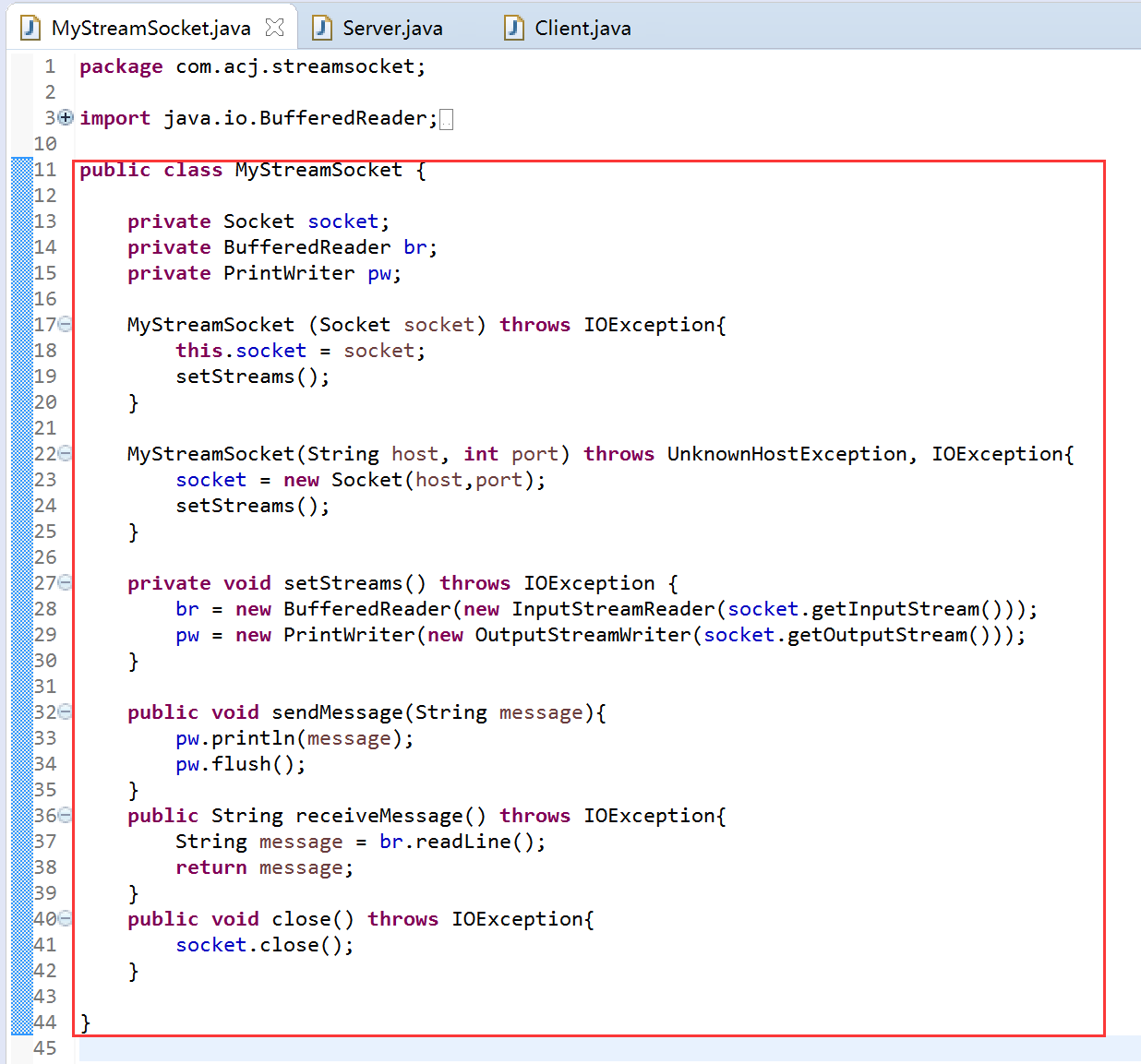


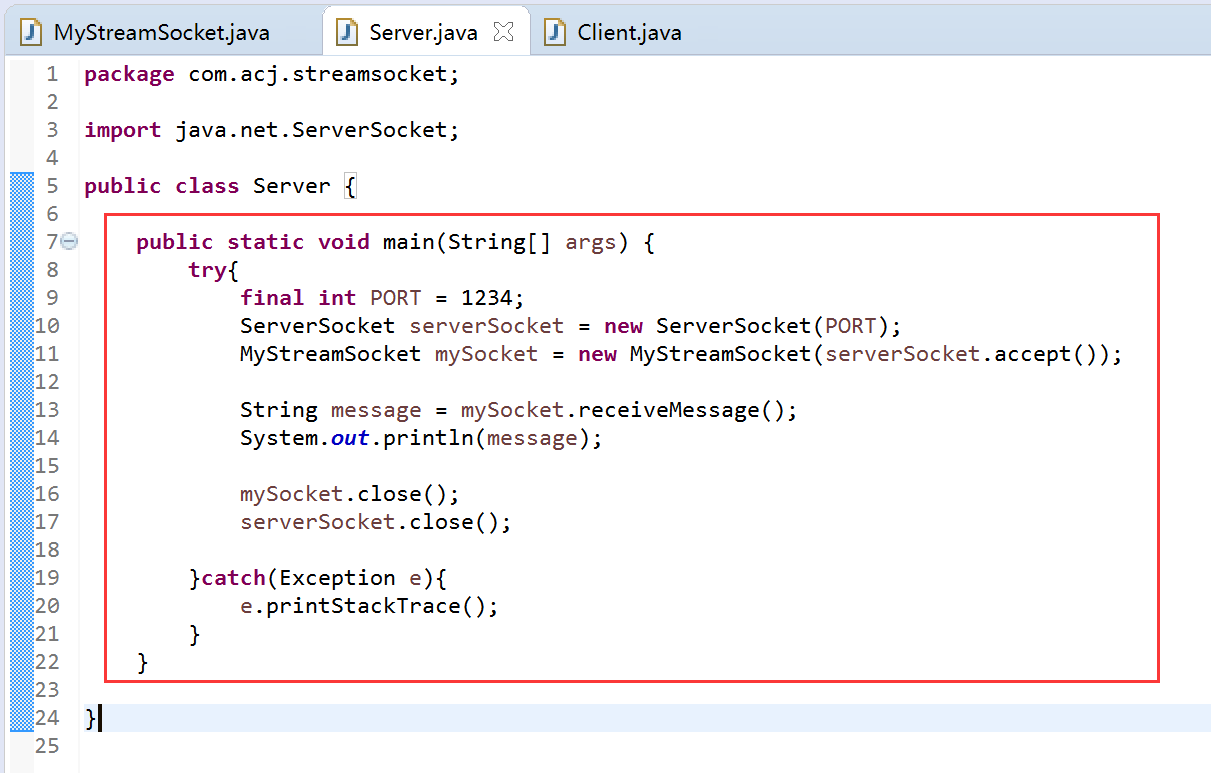
DatagramSocket 编程示例：

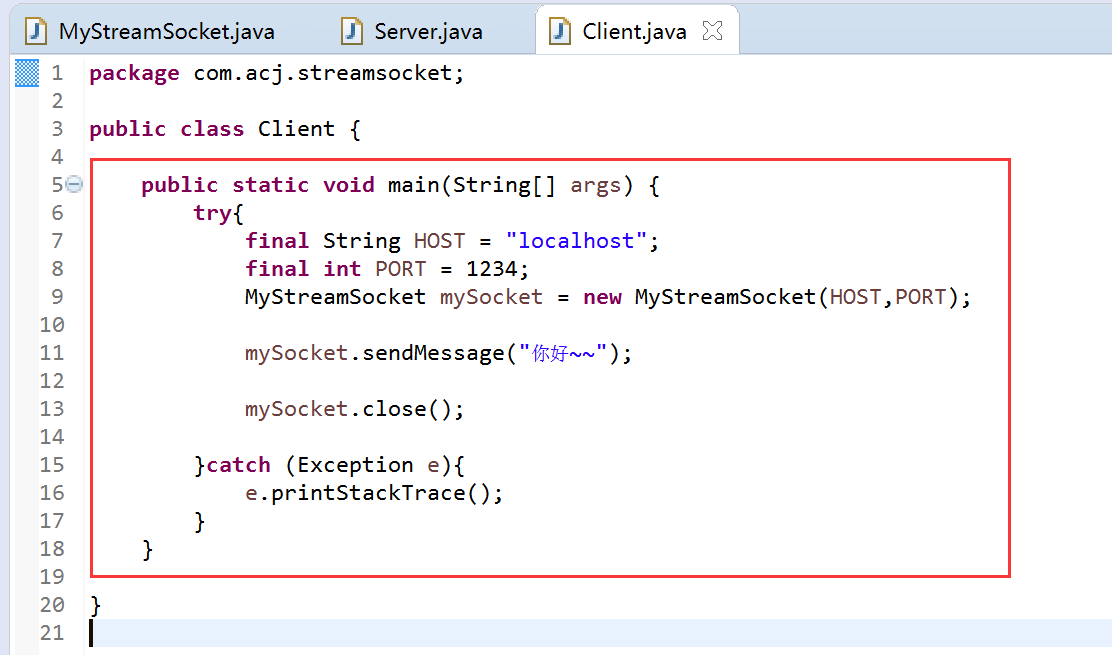




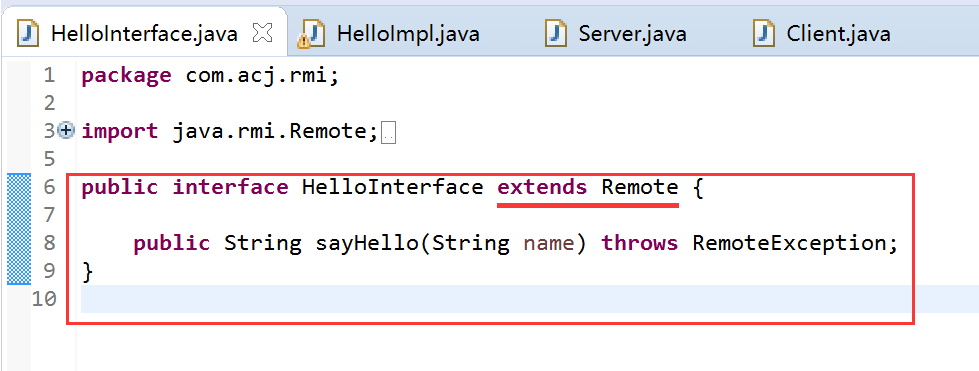
流式 Socket 编程示例：

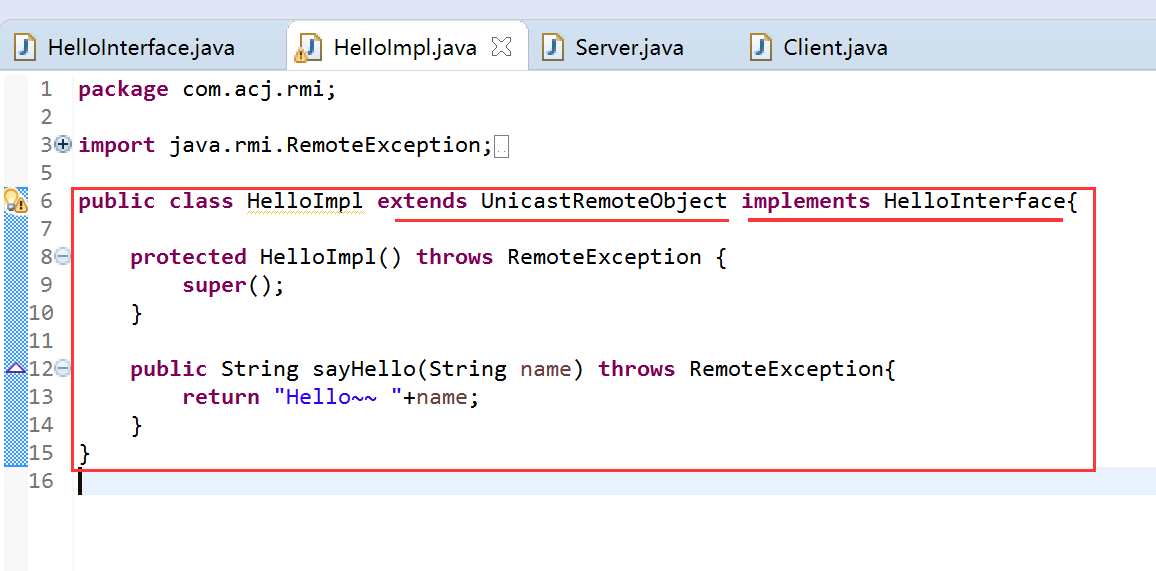


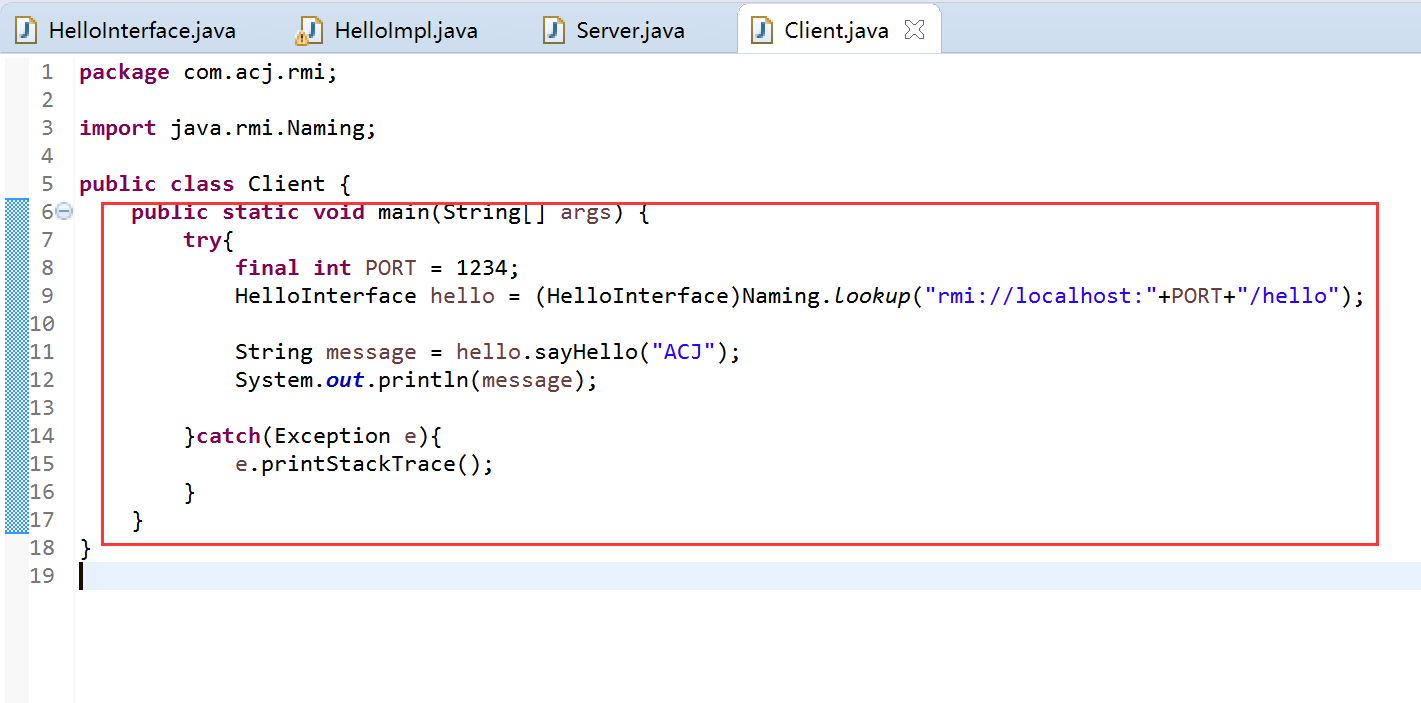




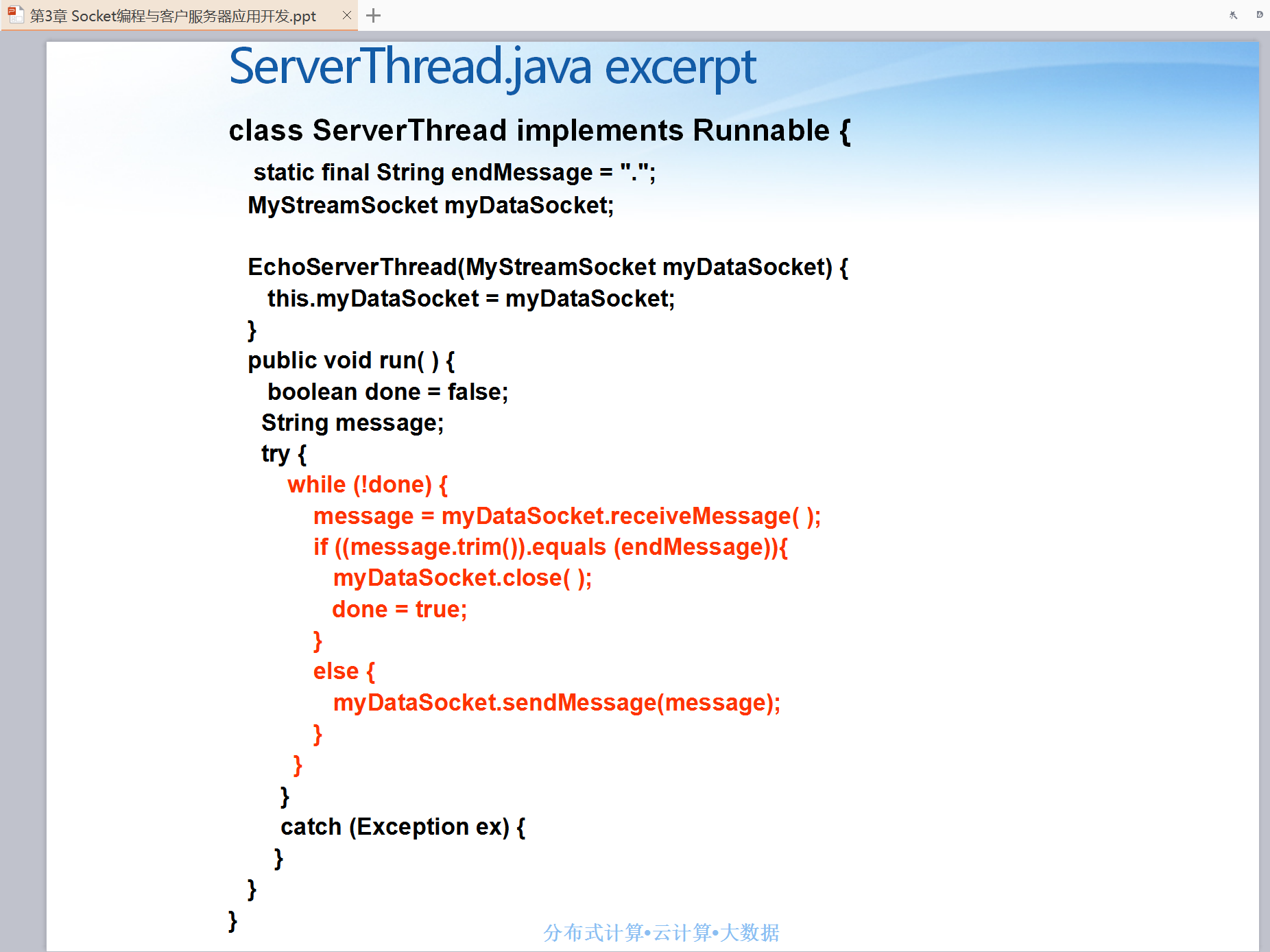
RMI 编程示例：







并发服务器原理：



# 五. 术语

**进程间通信：IPC， InterProcess Communication**

**CAP理论：**

**一致性： C， Consistency**

**可用性： A， Availability**

**分区容忍性： P， Partition**

**单播：unicast**

**组播：multicast**

**阻塞： blocking**

**非阻塞： unblocking**

**同步： synchronous**

**异步： asynchronous**

**消息传递范型：Message-passing paradigm**

**P2P范型: Peer-to-Peer paradigm**

**分布式对象范型：Distributed objects paradigm**

**消息系统范型： Message system models**

**点对点消息模型：Point-to-point message model**

**发布订阅消息模型：Public/Subscribe message model**

**远程过程调用： RPC， Remote Procedure Call**

**分布式对象范型：distributed objects paradigm**

**远程方法调用： RMI， Remote Method Invocation**

**对象请求代理： Object request Broker**

**网络服务范型： Network service paradigm**

**移动代理范型： Mobile agent paradigm**

**云服务范型：Cloud service paradigm**

**基础设施即服务：IaaS**

**平台即服务：PaaS**

**软件即服务：SaaS**

**UDP ： User Datagram Protocol**

**TCP ： Transmission Control Protocol**

**数据包Socket： DatagramSocket**

**流式Socket：Stream Socket**

**客户端回调：client callback,**

**桩下载：stub downloading**

**安全管理器：security m**anager