Java 服务器程序设计： 原理和技术

作者：Subrahmanyam Allamaraju

综述

建立和管理服务器端企业应用一直是挑战。在过去的二十年中，作用和服务器端的重要性应用有所增加。二十一世纪的经济决定了电子商务和其他企业应用是透明的，网络化，适应性强，服务至上。这些要求已大大改变了应用程序的性质。随着企业的迁移从封闭和孤立的应用程序更加透明，网络化，面向服务的应用，使电子业务交易，服务器端技术占据更加突出的位置。尽管在任何的应用程序的企业存在差距，服务器端应用程序是电力二十一世纪的企业！

本文的目的有两个方面。首先，本文试图以突出服务器端应用的技术需求，从而建立一套编程模型。其次，根据这些编程模型，本文介绍了到 Java 2 企业版（J2EE）的读者。如果你是一个服务器端编程初学者，或 J2EE，本文将帮助你获得一个什么样这项技术是对整体的观点。如果你是与一个或多个 J2EE 技术的熟悉，本文将提供基本的后面的服务器端编程的原则，并帮助您联系这些原则特定的 J2EE 技术。

简介

建立和管理服务器端企业应用一直是挑战。在过去的二十年中，作用和服务器端的重要性应用有所增加。二十一世纪的经济决定了电子商务和其他企业应用是透明的，网络化，适应性强，服务为本。这些要求已大大改变了应用程序的性质。

随着企业的迁移从封闭和孤立的应用程序更加透明，网络化，面向服务的应用，使电子业务交易，服务器端技术变得更加突出。尽管差距在任何的应用程序的企业的生存，服务器端应用程序是那些电力二十一世纪的企业。对于服务器端应用程序的历史可以追溯到大型机时代，一个时代 在此期间只大型机吼道跨企业。他们被集中在自然，所有的计算，从用户界面，到复杂的业务流程执行中，把交易，集中执行。此后推出的迷你合约，台式机，和关系数据库这一事件引起了客户的各种应用服务器风格。这有助于建立一个可以应用的方式转变，导致各建筑风格，如两，三年，多层次架构。在所有的样式，数据库为中心的客户机服务器体系结构是最广泛适应之一。其他客户机服务器应用形式包括：应用开发使用 remoteprocedure 调用（RPC），分布式组件技术，如通用对象，请求代理架构

（CORBA）和分布式组件对象模型 （DCOM）的，等等，正如这些技术和体系结构是多种多样的，因此也被为这类客户申请的性质。最常用的客户类型 这些应用程序是基于桌面（即那些使用 Visual Basic 或开发其他类似的语言）。其他类型的客户包括其他应用程序（即，如 企业应用集成），甚至 Web 服务器。 随后，我们今天看到的是互联网功能的服务器端的应用。该建设互联网基础技术功能的服务器端应用程序进化显着，而且更成熟，不太复杂的处理，你会后面看到这篇文章。除了提供更灵活的编程模型，当今的服务器端技术是基础设施密集。

规划模型

看完了两个可能的消息传递方法，现在让我们巩固规划要求，具体如下：

1。一个设置队列和主题的基础设施。

2。一种用于连接到消息基础设施和把或发布 API 消息。

3。基础设施的登记者/反对队列/主题订户的能力。

4。一个沉醉于基于消息队列或主题的 API。除了这些基本要求，下面的功能也被要求：

1。消息耐久度：基础设施应允许消息的耐久性（即是，一旦发布，消息不应该失去的，直到它被传递）。

2。保证交付：基础设施应保证交付。

3.一次性的交付：提供信息基础设施应该只有一次，一次。这是关键，因为重复的消息可能会产生不利影响业务由收件人实施进程。

4。为确认：基础设施也应该允许确认通知发送给发件人。我们会更详细地研究其中的一些功能，当我们讨论的 Java 消息服务（JMS）。

其他服务器端的要求

在上一节中，我们专注于编程方面是取决于客户如何和服务器相互通信。在本节中，我们将讨论两个最重要的基础设施有关的问题适用于企业级分布式事务的服务器端应用程序和安全性。分布式事务处理事务处理是至关重要的企业应用程序处理数据共享在各个业务流程。如果您熟悉基本的数据库编程，你可能也很熟悉与交易处理的基本概念。随着 JDBC40 时，的 setAutoCommit（布尔），提交（）和 rollback（）的方法 java.sql.Connection 接口让您提交或撤消的数据库组操作作为一个工作单元。或者，如果您开发客户服务器应用使用，也就是说，从甲骨文的快速应用开发的编程环境或微软，你应该使用的提交和回滚操作。做这些行动的目的是什么服务？您可能还记得，这些操作让你永久地记录（与提

交）或撤销（含回滚）所有数据库变化作出一个连接。例如，认为要实现一些数据库操作。使用提交/回滚操作，您可以完成或中止所有的变化。

该图显示了如何实现一个交易使用 JDBC 的 API。随着封闭的 setAutoCommit（假）和 commit（），所有的数据库操作演出之间被视为一个单一的工作单元。而不是调用犯（），您还可以调用回滚（），使得以前完成所有操作回滚（）将被撤销在数据库中。本的 setAutoCommit（false）和提交（）调用让您标记一个事务的界限。数据库使用的概念一个连接来实现这些交易。对于事务处理的基本要求之一是保存 ACID41 在数据库中的数据属性。酸的缩写代表 原子性，一致性，隔离性和持久性。这些属性意味着以下几点：

- 你应该能够将多个数据库操作成原子单位工作。
- 数据库操作应该是一致的，相对于完整的业务逻辑修改数据。
- 资料库由多个应用程序上执行相同的数据操作应不会互相影响。
- 数据库操作的效果应该是永久性的。虽然这些数据的属性可以维持今天的关系型数据库系统，数据库不能单独保存在服务器端的情况下，性能申请。是什么样的服务器端分布式应用程序的特别之处？下一个部分将介绍您给处理此类交易有关的问题分布式服务器端环境。为了了解是否需要一个分布式事务处理为分布式应用基础设施。

至于数据库服务器而言，有两个不同的连接，因此，两个数据库事务。但是，我们可以把两个独立的交易作为一个单一的工作单元的一部分？这个问题时似乎更多地参与各经营的对象是在不同的数据库。但是，需要什么来维持单在这些单位业务工作？怎样才能为我们提供的语义提交和回滚在这种情况下？在一个连接对象中定义的基本操作是不是就足够，因为我们有两个数据库连接问题。在为了实现跨这两个连接一个工作单元（即交易），都数据库操作集必须提交或回滚在一起。然而，两套操作发生在两个不同的数据库中连接，每一个都是本地的对象，得到了连接。协调行动，例如，无论提交或回滚两套在一起需要一个第三方，并使用以下方法：

1。客户端（未显示）调用对象方法 A

2。 Object 一个通知第三方，它将执行某些数据库操作。

3。对象 A 执行一些数据库操作。

4。对象 A 调用对象 B

5。对象 B 执行一些更多的数据库操作。

6。对象 B 的回报。

7。 Object 一个通知第三方一旦它已经完成了数据库操作。

8。第三方验证的数据库操作都可以被提交或回滚。

9。第三方指示数据库提交或回滚各自数据库操作。

10。对象的一个回报。列出的步骤，现在似乎更长，但在此之前，我们看看发生了什么，我们将讨论的第三方的角色。由于 A 和 B•只关心各自的数据库操作和可能是不知道所有的数据库操作被其他演出，我们需要第三方来监控整个两个对象的所有操作。

•由于 A 和 B 可以提交/回滚数据库操作的有关规定，我们需要一个第三方来协调跨两个单提交/回滚。这个第三方是所谓的，事务管理器或事务监视器，以及协调过程提交/跨越上述列出的对象回滚被调用，两相 commit.42 在这个过程中，事务管理器轮询数据库（S）以找到数据库操作是否可以提交。如果答案是肯定的，交易经理要求提交的业务。正如你可以看到，从前面的列表的步骤，对象 A 必须告知事务管理器，它即将进行交易，然后再启动数据库
业务。同样，在所有操作结束，则必须告知事务划分。它是通过它的边界是交易创建的过程业务。这些界限内进行的所有操作都被视为一个单一单位的工作。
注：除对象 A，从以前的例子中，客户还可以标定交易。
开放 43 组（以前的 X /公开组）中指定的第一个广泛使用交易处理模型，它是已知的分布式事务处理模型（DTP）的 0.44 该模型定义了一组 API，这是基本建立分布式事务的应用程序块。以下是两个的 API：

•XA 接口：定义在数据库和事务操作这样，数据库管理员可以注册自己的事务管理器和事务管理器可以与数据库进行交互，以协调事务。

•TX 接口：定义操作事务经理应实现这样的应用程序（对象 A 和 B 以上）可以划分交易和控制交易时 required.45 大多数数据库都支持 TX 接口和 XA 接口数据库的一部分。商业事务处理 systems46 如 Tuxedo，（从 BEA 系统公司）以及恩西纳（现在是 IBM 的 WebSphere 产品套件的一部分），实施的 TX 事务管理器接口。

另一个事务处理模型是对象管理集团（OMG 的）47 对象事务服务（OTS）.48 这种模式是用于分布式对象应用程序使用 CORBA，它规定了某些接口，支持分布在 CORBA 对象的交易。在 Java 领域，以下是两个 API 的分布式处理交易：

1。 Java 事务服务（JTS）49：这是 OTS 的 Java 映射，它是适合执行 Java 事务经理。

2。 Java 事务 API（JTA）的 50：这是一种高层次的交易为 Java API 服务器端应用程序，它为交易处理中使用 J2EE 的模型申请。这些技术提供交易管理编程方式（即客户/对象是事务划分的责任）。在 J2EE，以及微软事务服务器（MTS），

提供了另一种机制，即，声明式事务处理。使用此过程中，您可以启用事务处理某些配置，而不是通过实施一些交易的逻辑。安全在商业和技术界，安全是一种最常见的误用和误解的概念。尽管各方对安全的关注谁与企业应用开发和使用协议，安全是一个最不关心的要求。其中一个原因是因为安全漏洞可
发生下列情形之一：

1。网络 Security51：在这个级别的失误包括对违反网络服务器。这种违反解决方案包括防火墙，非军事区等。

2。操作系统 Security52：这是另一个的安全级别处理保护各种操作系统级别的资源（文件，线程，进程等）。一旦入侵者破坏了网络，操作系统的下一个目标访问操作系统级别的资源。

3。服务器端应用程序安全：防止未知或该级别的交易访问服务器端的应用程序未经授权的用户。

4。数据库安全性：除了以前的水平，数据库厂商也提供自己的安全机制来防止未经授权的访问数据。因为我们正在处理的服务器端应用程序在这一块，覆盖范围安全将是有限的安全机制在应用程序级别。在企业，安全问题出现时应用提供资源访问多个内部和外部，用户和客户端应用。至于安全方面，资源可以是业务数据（即购买订单或客户档案）各企业应用维护，也可以是的执行，或参与，业务流程由一个或多个实施企业应用程序。下面的列表显示了一些典型的安全关切的问题

企业应用程序：

•用户身份验证：当您的应用程序处理的特权业务流程和业务敏感数据，重要的是要建立一个用户/客户的身份，然后提供用户/客户端访问的数据或业务流程。验证过程有助于建立用户/客户的身份。认证通常包括用户/客户端交换一个令牌（一密码的应用程序）上，用户/客户端试图访问。对于这一机制，令牌都应该知道的用户/客户端和申请。 （更详细的机制，包括数字证书，生物识别，等）

•允许/禁止访问，授权：身份验证仅是不够的，尤其是当有不同类型的用户/客户提供不同的权限在一个给定的环境。在一个典型的企业，可能会遇到不同的用户根据不同的身份参与和角色不同的业务流程。只有这些用户/允许在某些业务流程或客户参与访问某些数据应被允许访问/做任务。所有其他用户/客户没有所需的身份或角色不应该提供访问。对授权进程（有时被称为访问控制）提供或拒绝访问，基于的身份/角色的用户。 （注之间的身份验证和差异授权。虽然与证明身份，授权认证协议与或拒绝提供基于身份的访问协议。）

•线级安全性的数据保密性和完整性：通过今天的互联网的网络应用，通过几个公共网络数据传播，提高对是否有人关注，应当允许记录/监视器或即使在传输过程中修改的数据。对数据的性质而定，如违规行为可能导致的后果是非常有害的业务。为了防止这一点，就必须保证数据的保密性和完整性。数据从开始到保密防止解密数据，而一个人过境。涉及数据完整性检查的引进，使通信双方（客户机和服务器）来检测任何类型的篡改数据。数字签名，各种加密 technologies53 和协议，如为安全套接字层（SSL），在维护数据的保密性和帮助完整性。以下为服务器端应用程序和编程的关键要求在这篇文章中讨论的模型

1。手段来验证用户和客户：一些用户认证方式与客户和维护整个应用程序的所有部分的身份。 （此要求持有）供客户基于 Web 和其他形式。

2。URL 访问控制:这是必需的,当服务器端应用程序可以达到了通过 HTTP。

3。控制方法调用：在远程方法调用控制对象。

4。消息队列访问控制：通过向消息队列和访问控制主题。

5。能够插入电线水平在技术：技术的能力，以堵塞在导线水平保证了数据的保密性和完整性。我们将看到这些是如何在 J2EE 应用程序实现的下一节。

J2EE 的服务器端编程

到这一点在本文中，我们讨论后的服务器端的各项原则计算。在讨论中，虽然我已提到的一些服务器端的 Java 技术，我故意忽略了这些细节，如何原则映射到不同的服务器端使用 Java 技术。从本节展望未来，我们将具体讨论在 J2EE 的 Java 服务器端技术。

J2EE 是一个综合的平台相结合的各种服务器端编程模型前面讨论这篇文章。Sun Microsystems 公司推出的 J2EE 下旬 199954，虽然一些 J2EE 的 API，比如 servlet，Java 命名和目录接口（JNDI）等，分别出台之前，Sun 微系统集成所有这些与连贯集成架构技术的服务器端编与 J2EE。今天，J2EE 是建立服务器端和互联网的事实标准使用 Java 应用程序，并通过一些厂商通过提供 J2EE 兼容应用服务器。J2EE 是一个规范，它本质上是如何规定各种 J2EE 技术一起工作。这些技术在各种讨论每一个由 Sun Microsystems 公布的规格。 Sun 维护一个引用实现 J2EE 的。尽管如此，仍然是一个 J2EE 的几个规格商业和开源实现。虽然这样的实现是通常被称为 J2EE 应用服务器，我们将改为使用术语的 J2EE 平台，以表示的 J2EE 技术的实施。 J2EE 的包括以下技术：

•企业 JavaBean（EJB）：用于开发分布式组件使用（http://java.sun.com/products/ejb）。

•Java Servlets 和 Java 服务器页面（JSP 页面）：用于建立基于 Web 的应用（http://java.sun.com/products/servlet，
http://java.sun.com/products/jsp）。

现在让我们讨论如何对这些需求的 J2EE 都提供。在我们进入细节如何去这些编程模型的支持，我们将着眼于最常用的 J2EE 的术语，如下一些：

1。集装箱：我们使用的术语来表示一个容器运行，支持应用程序。容器有两种类型的开发服务器端使用 J2EE 应用程序。EJB 容器：EJB 容器的存取服务器端功能可用于同步请求响应开发的对象模型。在通过 J2EE1.3 和 EJB2.0，即将增强的 EJB 容器也是托管服务器端对象的能力，可以通过异步消息被激活。Web 容器：Web 容器的对象是有能力举办开发了无状态请求响应模型通过 HTTP 通信。Web 容器维护实例对 Java servlet 和 JSP 页面。

2。应用组件：J2EE 规范使用术语应用组件来表示如 EJB，Java Servlet 的容器管理的对象，并 JSP 页面。这些都是管理，作为容器（即运行时）维护这些实例的生命周期。 （开发人员没有明确创建这些实例。）该容器还管理和安全等作为交易的事情，因为我们将看到在本文后面。

3。模块：除了应用组件的概念，采用了 J2EE 的概念的模块。一个模块由一个或多个特定类型的应用程序组件。例如，一个 EJB 模块由一个或多个的 EJB 组件，同样，一个 Web 模块由一个或多个 servlet 和 JSP 页（连同静态内容，如 HTML 文件，图像等）。主要一个模块的目的是为了让软包装结构。在 J2EE 中，模块打包成 Web 归档（WAR）或 Java 归档（JAR）文件。

4。 J2EE 应用程序：一个 J2EE 应用程序是一个或多个模块组合。J2EE 应用程序可以被打包到企业归档（EAR）文件。除了担任高层次应用模块封装的一种手段，一个 J2EE 应用程序还可以定义各种应用程序之间虚拟边界部署在一个给定的容器。

5。部署：这是另话，你会经常遇到在 J2EE 世界。虽然部署严格的定义已经超出了本文的范围，在简单来说，部署是添加到 J2EE 应用程序的过程平台。这可能意味着简单照搬类(适当打包成申请)进入到一个目录由 J2EE 平台的实现，也可能意味着使用供应商特定的部署工具来添加一个 J2EE 应用程序平台。一旦部署，J2EE 应用组件可用于通过对客户的要求为基础的 J2EE 容器调用（即，从 Web 请求浏览器或其他客户端），甚至其他 J2EE 应用程序组件。

6。部署描述：简单来说，是一个部署描述符使用 XML 配置文件中表示。随着应用程序组件,每个模块包括一个部署描述符。根据应用程序的类型组成部分，包括部署描述符的具体配置信息。部署描述符容器不仅帮助破译模块的内容，但

也让容器管理的各种生命周期应用程序组件，交易，甚至安全。