

JXTAPlatformJAVA参考实现源代码分析系列文章(2) PDF转换可能丢失图片或格式, 建议阅读原文

https://www.100test.com/kao_ti2020/252/2021_2022_JXTAPlatfo_c104_252458.htm

1 引言 管道的概念源于Unix,是不同线程之间直接传输数据的基本手段。JDK中java.io包中就有管道类,同时,管道在JXTA中是最基本的概念,是对等点之间的数据传输的主要方式。对等管道协议(PBP)明确规范了对等管道的绑定,解析,响应。本文依次剖析集中式(JDK)和对等环境下(JXTA)管道的实现方式,对比分析其异同,然后尝试在JXTA中建立一个虚拟的全双工的管道。本文的目标是通过对不同环境下管道的实现方式对比分析,来理解为什么JXTA采用管道作为基本的数据传输手段。

2 管道的形象化描述 一个生活中的情景:现在有两个地区A,B。A是石油生产区,B是石油消费区,现在B地区需要消费A地区的石油,当然可以通过海运,空运获得,然而最通常的方式是架设输油管道。如图所示:

1 引言 管道的概念源于Unix,是不同线程之间直接传输数据的基本手段。JDK中java.io包中就有管道类,同时,管道在JXTA中是最基本的概念,是对等点之间的数据传输的主要方式。对等管道协议(PBP)明确规范了对等管道的绑定,解析,响应。本文依次剖析集中式(JDK)和对等环境下(JXTA)管道的实现方式,对比分析其异同,然后尝试在JXTA中建立一个虚拟的全双工的管道。本文的目标是通过对不同环境下管道的实现方式对比分析,来理解为什么JXTA采用管道作为基本的数据传输手段。

2 管道的形象化描述 一个生活中的情景:现在有两个地区A,B。A是石油生产区,B是石油消费区,现在B地区需要消费A地区的石油,当然可以

通过海运，空运获得，然而最通常的方式是架设输油管道。如图所示：java中流的概念和管道的概念都可以通过此案例阐述，A与B之间连接的就是管道，负责将A的石油向B输出。A向管道输出数据(output)，B从管道输入数据(input)，可以这样理解，管道是A的输出对象，是B的数据源。这里就产生了三个类：输出流A，输入流B，管道。输入流B负责如何获取数据(read操作)，输出流A负责如何消费数据(write操作),管道负责连接它们(connect操作)。其实，在实现时，管道类分解为管道口，管道出口，由入口出口负责连接。在复杂的网络环境中，这种连接方式可以有专门的网络协议负责（例如，JXTA中的PBP,全称Pipe Bind Protocol）。由以上描述，我们可以清楚知道最原始的管道就是单向的，文章后面介绍的双向管道，是用两个单向管道虚拟的，而非真实的连接方式。不难发现管道最关键的问题是如何协调输出(A)与输入(B)。这在不同的网络环境会遇到不同的问题，最简单的是同一JVM下的不同过程(线程或任务)之间用同步方式传递数据。而对等环境下，如何去发现对方就是一个很现实的问题，这仅仅只是问题的其中之一，下面的章节会依次分析。

3 集中式环境下管道的实现

问题的描述：A与B是在同一JVM中，A,B有一方能够发现另一方的存在，A将数据发往B方，A发送数据与B接收数据是相互独立的。现在回到问题的最初：为什么要使用管道？A只管发送，B只管接受，那么数据在哪儿呢？经过下面的分析，就会明白管道把管理数据缓冲区的重任交给了他自己，A,B均是围绕这个缓冲区来启停线程的，显然这才是问题的本质。JDK中，类PipeInputStream（即前面所述的B）与PipeOutputStream（即前面所述的A）可以很好

的解决这一问题。首先给出类图如下。 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com