

<http://wiki.jikexueyuan.com/project/java-nio-zh/java-nio-overview.html>

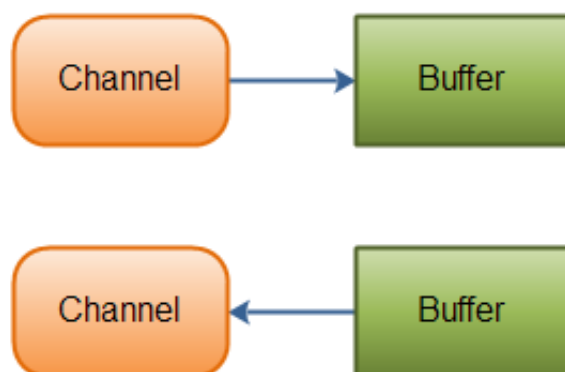
NIO包含下面几个核心的组件：

- Channels
- Buffers
- Selectors

整个NIO体系包含的类远远不止这几个，但是在笔者看来Channel, Buffer和Selector组成了这个核心的API。其他的一些组件，比如Pipe和FileLock仅仅只作为上述三个的负责类。因此在概览这一节中，会重点关注这三个概念。其他的组件会在各自的部分单独介绍。

## 通道和缓冲区（Channels and Buffers）

通常来说NIO中的所有IO都是从Channel开始的。Channel和流有点类似。通过Channel，我们即可以从Channel把数据读到Buffer中，也可以把数据冲Buffer写入到Channel，下图是一个示意图：



### Java NIO: Channels read data into Buffers, and Buffers write data into Channels

有很多的Channel， Buffer类型。下面列举了主要的几种：

- FileChannel -----File
- DatagramChannel -----UDP
- SocketChannel -----TCP
- ServerSocketChannel -----TCP

正如你看到的，这些channel基于于UPD和TCP的网络IO，以及文件IO。 和这些类

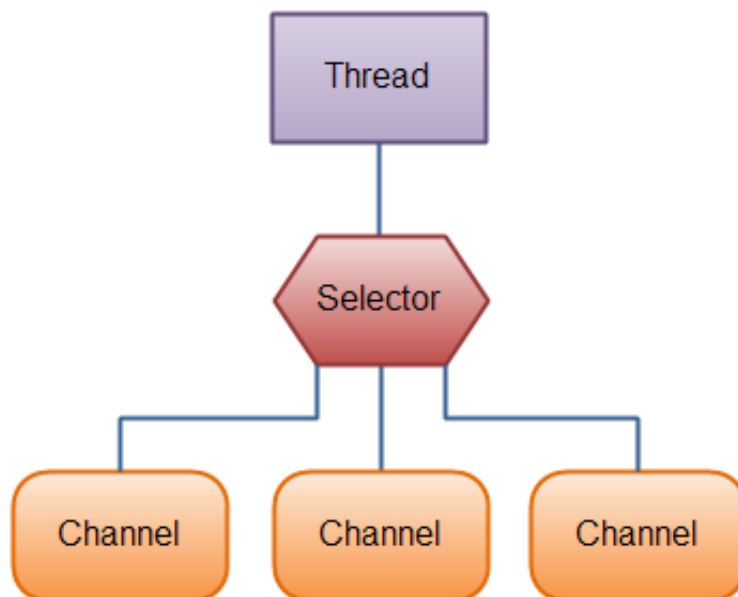
一起的还有其它一些比较有趣的接口，在本节中暂时不多介绍。为了简介起见，我们会在必要的时候引入这些概念。下面是核心的Buffer实现类的列表：

- ByteBuffer
- CharBuffer
- DoubleBuffer
- FloatBuffer
- IntBuffer
- LongBuffer
- ShortBuffer

这些Buffer涵盖了可以通过IO操作的基础类型：byte,short,int,long,float,double以及characters. NIO实际上还包含一种MappedByteBuffer,一般用于和内存映射的文件。

## 选择器（Selectors）

选择器允许单线程操作多个通道。如果你的程序中有大量的链接，同时每个链接的IO带宽不高的话，这个特性将会非常有帮助。比如聊天服务器。下面是一个单线程中Selector维护3个Channel的示意图：



### Java NIO: A Thread uses a Selector to handle 3 Channel's

要使用Selector的话，我们必须把Channel注册到Selector上，然后就可以调用Selector的select()方法。这个方法会进入阻塞，直到有一个channel的状态符合条

件。当方法返回后，线程可以处理这些时间。