

ImportNew

- [首页](#)
- [所有文章](#)
- [资讯](#)
- [Web](#)
- [架构](#)
- [基础技术](#)
- [书籍](#)
- [教程](#)
- [Java小组](#)
- [工具资源](#)

- 导航条 - ▼

Java NIO系列教程（4）：Scatter/Gather

2016/03/30 | 分类：[教程](#) | [0 条评论](#) | 标签：[Java NIO](#), [Scatter/Gather](#)

分享到：

20 译文出处：[郭董](#) 原文出处：[Jakob Jenkov](#)

Java NIO开始支持scatter/gather，scatter/gather用于描述从Channel（译者注：Channel在中文经常翻译为通道）中读取或者写入到Channel的操作。

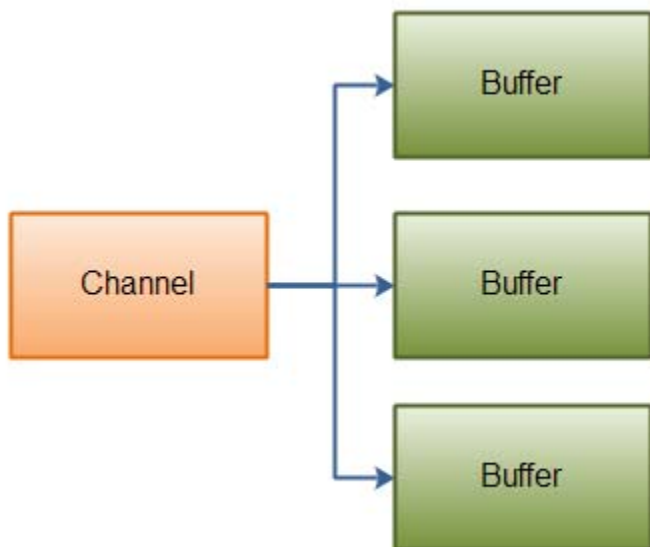
分散（scatter）从Channel中读取是指在读操作时将读取的数据写入多个buffer中。因此，Channel将从Channel中读取的数据“分散（scatter）”到多个Buffer中。

聚集（gather）写入Channel是指在写操作时将多个buffer的数据写入同一个Channel，因此，Channel将多个Buffer中的数据“聚集（gather）”后发送到Channel。

scatter / gather经常用于需要将传输的数据分开处理的场合，例如传输一个由消息头和消息体组成的消息，你可能会将消息体和消息头分散到不同的buffer中，这样你可以方便的处理消息头和消息体。

Scattering Reads

Scattering Reads是指数据从一个channel读取到多个buffer中。如下图描述：



Java NIO: Scattering Read

代码示例如下：

```
1 ByteBuffer header = ByteBuffer.allocate(128);
2 ByteBuffer body   = ByteBuffer.allocate(1024);
3
4 ByteBuffer[] bufferArray = { header, body };
5
6 channel.read(bufferArray);
```

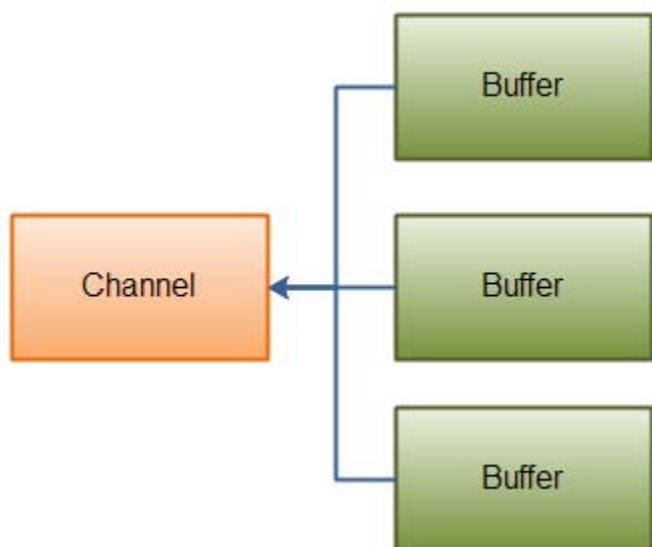
注意buffer首先被插入到数组，然后再将数组作为channel.read() 的输入参数。read()方法按照buffer在数组中的顺序将从channel中读取的数据写入到buffer，当一个buffer被写满后，channel紧接着向另一个buffer中写。



Scattering Reads在移动下一个buffer前，必须填满当前的buffer，这也意味着它不适用于动态消息(译者注：消息大小不固定)。换句话说，如果存在消息头和消息体，消息头必须完成填充（例如128byte），Scattering Reads才能正常工作。

Gathering Writes

Gathering Writes是指数据从多个buffer写入到同一个channel。如下图描述：



Java NIO: Gathering Write

代码示例如下：

```
1 ByteBuffer header = ByteBuffer.allocate(128);
2 ByteBuffer body   = ByteBuffer.allocate(1024);
3
4 //write data into buffers
5
6 ByteBuffer[] bufferArray = { header, body };
7
8 channel.write(bufferArray);
```

buffers数组是write()方法的入参，write()方法会按照buffer在数组中的顺序，将数据写入到channel，注意只有position和limit之间的数据才会被写入。因此，如果一个buffer的容量为128byte，但是仅仅包含58byte的数据，那么这58byte的数据将被写入到channel中。因此与Scattering Reads相反，Gathering Writes能较好的处理动态消息。

本系列：

- [Java NIO系列教程（1）：Java NIO 概述](#)
- [Java NIO系列教程（2）：Channel](#)
- [Java NIO系列教程（3）：Buffer](#)
- [Java NIO系列教程（4）：Scatter/Gather](#)

20



相关文章

- [攻破JAVA NIO技术壁垒](#)
- [Java NIO系列教程（12）：Java NIO与IO](#)
- [Java NIO系列教程（11）：Pipe](#)
- [Java NIO系列教程（10）：Java NIO DatagramChannel](#)
- [Java NIO系列教程（9）：ServerSocketChannel](#)
- [Java NIO系列教程（8）：SocketChannel](#)
- [Java NIO系列教程（7）：FileChannel](#)
- [Java NIO系列教程（6）：Selector](#)
- [Java NIO系列教程（5）：通道之间的数据传输](#)
- [Java NIO系列教程（3）：Buffer](#)

发表评论

Comment form

Name*

姓名

邮箱*

网站 (请以 http://开头)

评论内容*

请填写评论内容

(*) 表示必填项

[提交评论](#)

还没有评论。

[« Java NIO系列教程（3）：Buffer](#)



[Java NIO系列教程（5）：通道之间的数据传输 »](#)

Search for:



- [本周热门文章](#)
- [本月热门](#)
- [热门标签](#)

0 [记一次集群内无可用 http 服务问题...](#)

1 [Java 技术之垃圾回收机制](#)

2 [公司编程竞赛之最长路径问题](#)

3 [Java 中的十个"单行代码编程" \(O...](#)

- 4 [Java 中 9 个处理 Exception ...](#)
- 5 [HttpClient 以及 Json 传递的...](#)
- 6 [浅析 Spring 中的事件驱动机制](#)
- 7 [浅析分布式下的事件驱动机制（ PubS...](#)
- 8 [探索各种随机函数（ Java 环境...](#)
- 9 [Java 守护线程概述](#)



最新评论

- 
Re: [攻破JAVA NIO技术壁垒](#)
Hi, 请到伯乐在线的小组发帖提问, 支持微信登录。链接是: <http://group.jobbole....> 唐尤华
- 
Re: [攻破JAVA NIO技术壁垒](#) 
TCP服务端的NIO写法 服务端怎么发送呢。原谅小白 菜鸟
- 
Re: [关于 Java 中的 double check ...](#)
volatile 可以避免指令重排啊。所以double check还是可以用的。 hipilee
- 
Re: [Spring4 + Spring MVC + M...](#)
Hi, 请到伯乐在线的小组发帖提问, 支持微信登录。链接是: <http://group.jobbole....> 唐尤华
- 
Re: [Spring4 + Spring MVC + M...](#)
我一直不太明白, spring的bean容器和springmvc的bean容器之间的关系。 hw_绝影
- 
Re: [Spirng+SpringMVC+Maven+Myba...](#)
很好, 按照步骤, 已经成功。 莫凡
- 
Re: [Spring中@Transactional事务...](#)
声明式事务可以用aop来实现, 分别是jdk代理和cglib代理, 基于接口和普通类. 在同一个类中一个方... chengjiliang



Re: [关于 Java 中的 double check ...](#)

在JDK1.5之后，用volatile关键字修饰_INSTANCE属性 就能避免因指令重排导致的对象...
Byron

关于ImportNew

ImportNew 专注于 Java 技术分享。于2012年11月11日 11:11正式上线。是的，这是一个很特别的时刻：)

ImportNew 由两个 Java 关键字 import 和 new 组成，意指：Java 开发者学习新知识的网站。import 可认为是学习和吸收，new 则可认为是新知识、新技术圈子和新朋友.....



联系我们

Email : ImportNew.com@gmail.com

新浪微博：[@ImportNew](#)

推荐微信号



ImportNew



安卓应用频道



Linux爱好者



反馈建议：ImportNew.com@gmail.com

广告与商务合作QQ：2302462408

推荐关注

[小组](#) – 好的话题、有启发的回复、值得信赖的圈子

[头条](#) – 写了文章？看干货？去头条！

[相亲](#) – 为IT单身男女服务的征婚传播平台

[资源](#) – 优秀的工具资源导航

[翻译](#) – 活跃 & 专业的翻译小组

[博客](#) – 国内外的精选博客文章

[设计](#) – UI,网页，交互和用户体验

[前端](#) – JavaScript, HTML5, CSS

[安卓](#) – 专注Android技术分享

[iOS](#) – 专注iOS技术分享

[Java](#) – 专注Java技术分享

[Python](#) – 专注Python技术分享

© 2017 ImportNew