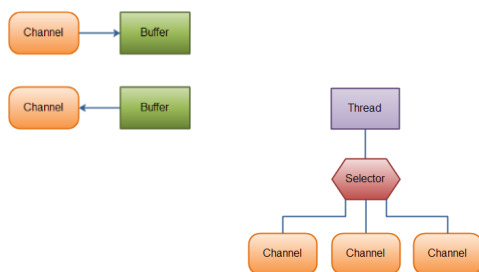## JAVA NIO

Veerle Ongenae

Java NIO

- Java New IO
- Alternatief
  - Java IO
  - Java Networking

Industrieel Ingenieur Informatica, UGent

Basisconcepten

Industrieel Ingenieur Informatica, UGent
http://tutorials.jenkov.com/java-nio/index.html
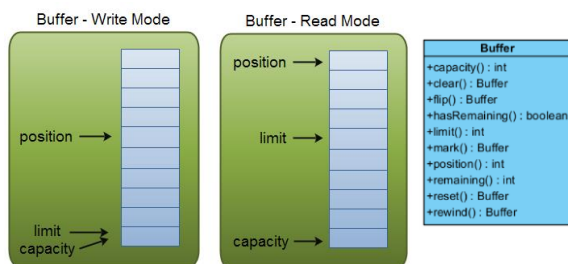

Channel

```
RandomAccessFile aFile = new RandomAccessFile("data/nio-data.txt",
"rw");
FileChannel inChannel = aFile.getChannel();
ByteBuffer buf = ByteBuffer.allocate(48);
int bytesRead = inChannel.read(buf);
while (bytesRead != -1) {
  System.out.println("Read " + bytesRead);
  buf.flip();
  while(buf.hasRemaining()){
    System.out.print((char) buf.get());
  }
  buf.clear();
  bytesRead = inChannel.read(buf);
}
aFile.close();
```

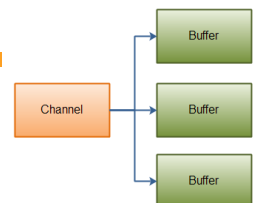Industrieel Ingenieur Informatica, UGent

---

Buffer

Industrieel Ingenieur Informatica, UGent
http://tutorials.jenkov.com/java-nio/index.html

Scattering reads


```
ByteBuffer header = ByteBuffer.allocate(128);
ByteBuffer body   = ByteBuffer.allocate(1024);

ByteBuffer[] bufferArray = { header, body };

channel.read(bufferArray);
```

Industrieel Ingenieur Informatica, UGent
http://tutorials.jenkov.com/java-nio/index.html

Gathering writes


```
ByteBuffer header = ByteBuffer.allocate(128);
ByteBuffer body   = ByteBuffer.allocate(1024);

//write data into buffers

ByteBuffer[] bufferArray = { header, body };

channel.write(bufferArray);
```
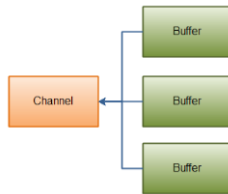Industrieel Ingenieur Informatica, UGent
http://tutorials.jenkov.com/java-nio/index.html

Channel to channel transfer

```
RandomAccessFile fromFile
  = new RandomAccessFile("fromFile.txt", "rw");
FileChannel fromChannel = fromFile.getChannel();

RandomAccessFile toFile
  = new RandomAccessFile("toFile.txt", "rw");
FileChannel toChannel = toFile.getChannel();

long position = 0;
long count = fromChannel.size();

toChannel.transferFrom(fromChannel, position, count);
```
Industrieel Ingenieur Informatica, UGent

Channel to channel transfer

```
RandomAccessFile fromFile
  = new RandomAccessFile("fromFile.txt", "rw");
FileChannel fromChannel = fromFile.getChannel();

RandomAccessFile toFile
  = new RandomAccessFile("toFile.txt", "rw");
FileChannel toChannel = toFile.getChannel();

long position = 0;
long count = fromChannel.size();

fromChannel.transferTo(position, count, toChannel);
```
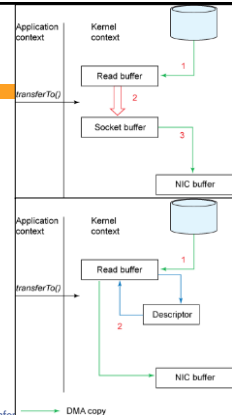Industrieel Ingenieur Informatica, UGent

Zero copy
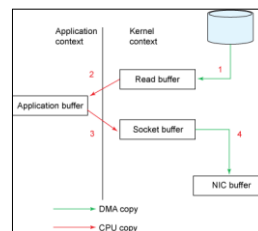
Industrieel Ingenieur Informatica, UGent
http://www.ibm.com/developerworks/library/j-zerocopy/

Selector

Industrieel Ingenieur Informatica, UGent
http://tutorials.jenkov.com/java-nio/index.html
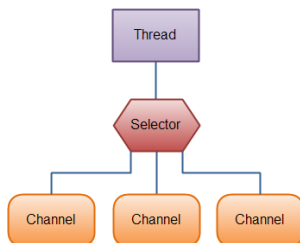
Selector – aanmaak en registratie

```
// aanmaak selector
Selector selector = Selector.open();

// registratie één of meerdere kanalen
channel.configureBlocking(false);
SelectionKey key
  = channel.register(selector, SelectionKey.OP_READ);
```
Industrieel Ingenieur Informatica, UGent

Selector - gebruik

`13`

```
while(true) {
  // wachten
  int readyChannels = selector.select();
  if(readyChannels == 0) continue;

  for (SelectionKey key : selector.selectedKeys()) {
    .. // kanaal klaar om te …
  }
  selector.selectedKeys().clear();
}
```

Industrieel Ingenieur Informatica, UGent

Selector - communicatie

`14`

```
if(key.isAcceptable()) {
    // a connection was accepted by a ServerSocketChannel.
    Channel channel = key.channel(); …
} else if (key.isConnectable()) {
    // a connection was established with a remote server.
} else if (key.isReadable()) {
    // a channel is ready for reading
} else if (key.isWritable()) {
    // a channel is ready for writing
}
…
```

Industrieel Ingenieur Informatica, UGent

SocketChannel: aanmaak - lezen

`15`

```
// aanmaken
SocketChannel socketChannel = SocketChannel.open();
socketChannel.connect(
  new InetSocketAddress("http://jenkov.com", 80));

// lezen
ByteBuffer buf = ByteBuffer.allocate(48);
int bytesRead = socketChannel.read(buf);
```

Industrieel Ingenieur Informatica, UGent

SocketChannel: schrijven - sluiten

`16`

```
// buffer met info
String newData = "New String to write to file...";
ByteBuffer buf = ByteBuffer.allocate(48);
buf.clear();
buf.put(newData.getBytes());
// data schrijven
buf.flip();
while(buf.hasRemaining()) {
    socketChannel.write(buf);
}
// afsluiten
socketChannel.close();
```

Industrieel Ingenieur Informatica, UGent

ServerSocketChannel: aanmaak

`17`

```
ServerSocketChannel serverSocketChannel =
  ServerSocketChannel.open();

serverSocketChannel.socket().bind(new InetSocketAddress(9999));

while(true){
  SocketChannel socketChannel = serverSocketChannel.accept();

  //do something with socketChannel...
}

serverSocketChannel.close();
```

Industrieel Ingenieur Informatica, UGent

Voorbeeld

`18`

☐ ChatServerProgram.java
☐ ChatServer.java

Industrieel Ingenieur Informatica, UGent

3

Java IO versus java NIO

19

IO	NIO
Stream oriented	Buffer oriented
Blocking IO	Non blocking IO
	Selectors

Industrieel Ingenieur Informatica, UGent

Java IO versus java NIO

20

Stream — Thread — Channel — Buffer — Thread

read data, block until ready
read data, block until ready
read data, block until ready
read data, block until ready

read data into buffer
Fill data into buffer
Check data in buffer
Goto top

Industrieel Ingenieur Informatica, UGent

Java IO versus java NIO

21

Connection / Thread / Server Socket / Connection / Thread

Industrieel Ingenieur Informatica, UGent

Java IO versus java NIO

22

- Change notifiers
- Selectors help multiplex
- Channels
- Memory mapping
- Character encoding and searching

Industrieel Ingenieur Informatica, UGent
http://www.javaworld.com/article/2078654/java-se/java-se-five-ways-to-maximize-java-nio-and-nio-2.html