FactorialClient Overview

What Happens

The factorial application uses the FactorialClient as the client driver and the FactorialServer as the server driver. Start the server first and then start the client.

Basically the application works with channels, pipelines and handlers. A channel uses a pipeline to process message events and a pipeline is composed of upstream (incoming) and downstream (outgoing) message event handlers. (An event handler can be both an upstream and a downstream handler.) A bootstrap class is used to merge the functionalities of the pipeline and the channel.

The input parameters are the host name of the server, the port of the server, and a number. The application calculates the factorial of the number. We have arbitrarily chosen this number to be the integer 5.

First we create a ChannelFactory that will return a NioClientSocketChannel, viz. the NioClientSocketChannelFactory. Second, we initialize the ClientBootstrap with the NioClientSocketChannelFactory. Third, we create a FactorialClientPipelineFactory that takes the number as a parameter. Then, fourth, we initialize the client bootstrap class with the pipeline. When the method getPipeline is called on this factory, it returns the FactorialClientPipeline initialized with the message event handlers: BigIntegerDecoder, NumberEncoder, and FactorialClientHandler. Fifth, the channel bootstrap sets the channel options to tcpNoDelay and keepAlive.

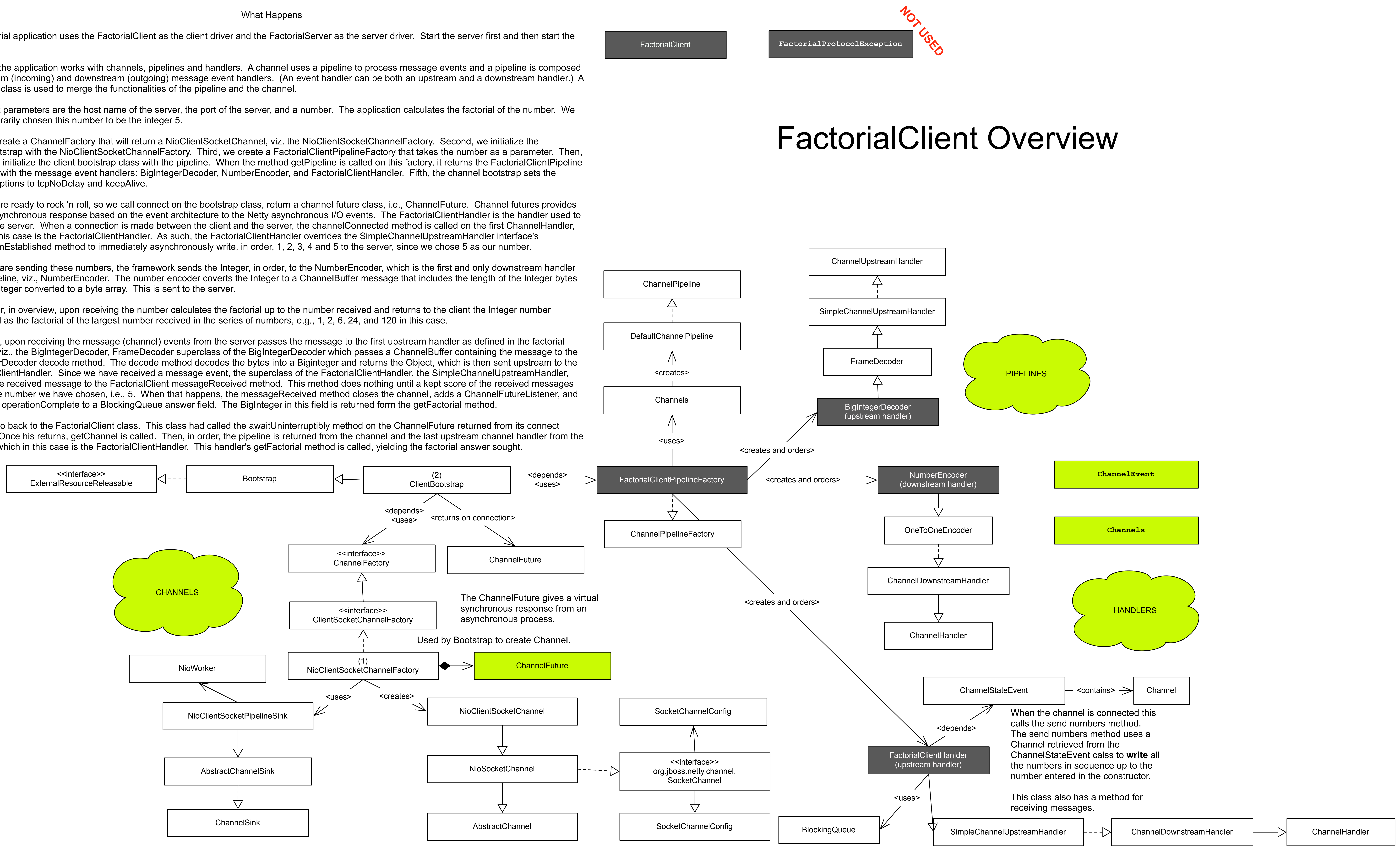
Now we are ready to rock 'n roll, so we call connect on the bootstrap class, return a channel future class, i.e., ChannelFuture. Channel futures provides a virtual synchronous response based on the event architecture to the Netty asynchronous I/O events. The FactorialClientHandler is the handler used to write to the server. When a connection is made between the client and the server, the channelConnected method is called on the first ChannelHandler, which in this case is the FactorialClientHandler. As such, the FactorialClientHandler overrides the SimpleChannelUpstreamHandler interface's connectionEstablished method to immediately asynchronously write, in order, 1, 2, 3, 4 and 5 to the server, since we chose 5 as our number.

Since we are sending these numbers, the framework sends the Integer, in order, to the NumberEncoder, which is the first and only downstream handler in the pipeline, viz., NumberEncoder. The number encoder coverts the Integer to a ChannelBuffer message that includes the length of the Integer bytes and the Integer converted to a byte array. This is sent to the server.

The server, in overview, upon receiving the number calculates the factorial up to the number received and returns to the client the Integer number calculated as the factorial of the largest number received in the series of numbers, e.g., 1, 2, 6, 24, and 120 in this case.

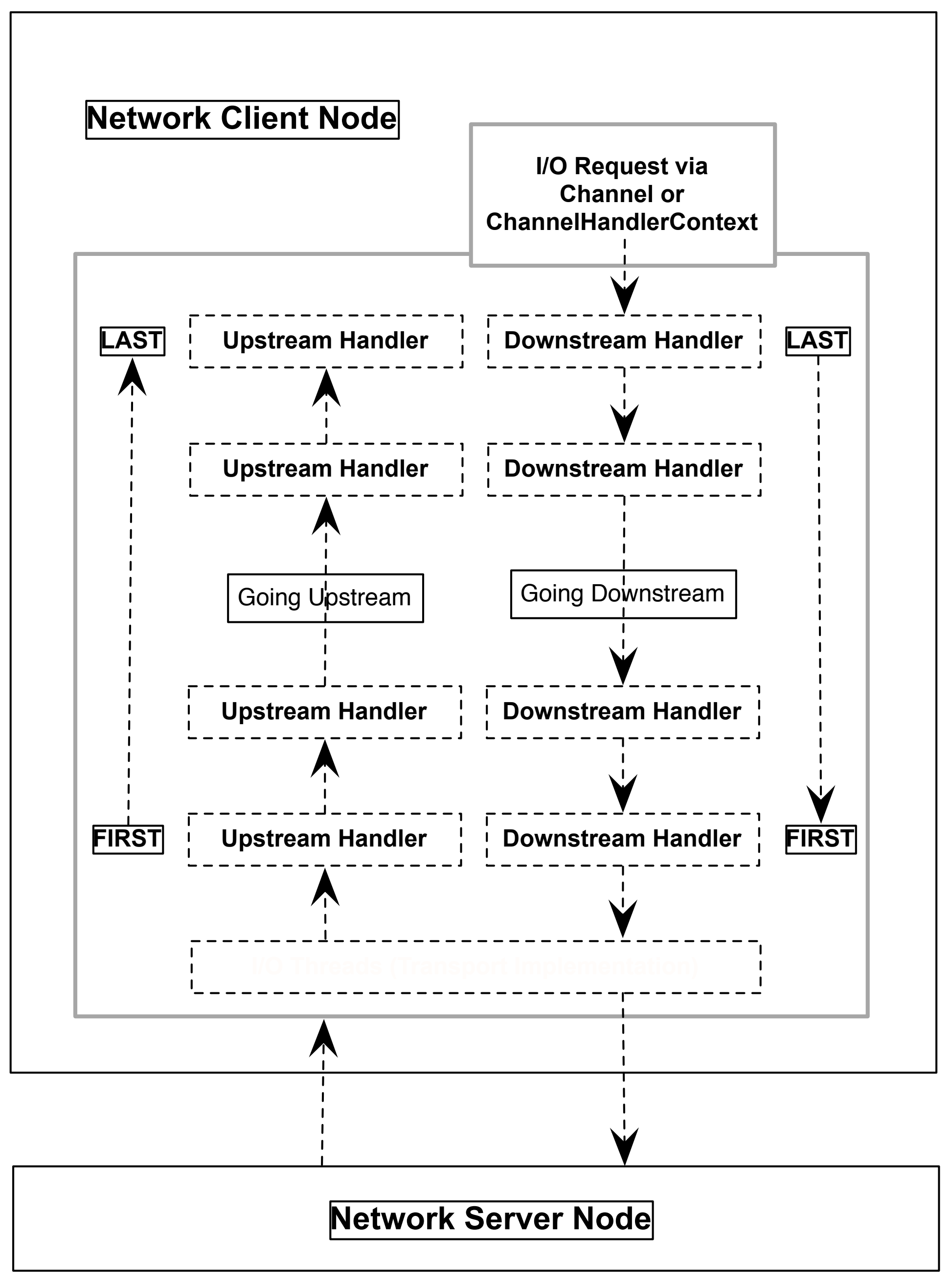
The client, upon receiving the message (channel) events from the server passes the message to the first upstream handler as defined in the factorial pipeline, viz., the BigIntegerDecoder, FrameDecoder superclass of the BigIntegerDecoder which passes a ChannelBuffer containing the message to the BigIntegerDecoder decode method. The decode method decodes the bytes into a Biginteger and returns the Object, which is then sent upstream to the FactorialClientHandler. Since we have received a message event, the superclass of the FactorialClientHandler, the SimpleChannelUpstreamHandler, passes the received message to the FactorialClient messageReceived method. This method does nothing until a kept score of the received messages equals the number we have chosen, i.e., 5. When that happens, the messageReceived method closes the channel, adds a ChannelFutureListener, and writes the operationComplete to a BlockingQueue answer field. The BigInteger in this field is returned form the getFactorial method.

Now we go back to the FactorialClient class. This class had called the awaitUninterruptibly method on the ChannelFuture returned from its connect method. Once his returns, getChannel is called. Then, in order, the pipeline is returned from the channel and the last upstream channel handler from the pipeline, which in this case is the FactorialClientHandler. This handler's getFactorial method is called, yielding the factorial answer sought.


FactorialClient
FactorialProtocolException
NOT USED
ChannelPipeline
DefaultChannelPipeline
<creates>
Channels
<uses>
ChannelUpstreamHandler
SimpleChannelUpstreamHandler
FrameDecoder
BigIntegerDecoder (upstream handler)
PIPELINES
<creates and orders>
<<interface>> ExternalResourceReleasable
Bootstrap
(2) ClientBootstrap
<depends> <uses>
FactorialClientPipelineFactory
<creates and orders>
NumberEncoder (downstream handler)
ChannelEvent
Channels
ChannelPipelineFactory
OneToOneEncoder
ChannelDownstreamHandler
ChannelHandler
HANDLERS
<depends> <uses>
<returns on connection>
<<interface>> ChannelFactory
ChannelFuture
CHANNELS
<<interface>> ClientSocketChannelFactory
The ChannelFuture gives a virtual synchronous response from an asynchronous process.
Used by Bootstrap to create Channel.
<creates and orders>
NioWorker
(1) NioClientSocketChannelFactory
ChannelFuture
<uses>
<creates>
NioClientSocketPipelineSink
NioClientSocketChannel
SocketChannelConfig
ChannelStateEvent
<contains>
Channel
<depends>
When the channel is connected this calls the send numbers method. The send numbers method uses a Channel retrieved from the ChannelStateEvent calss to write all the numbers in sequence up to the number entered in the constructor.
This class also has a method for receiving messages.
AbstractChannelSink
NioSocketChannel
<<interface>> org.jboss.netty.channel. SocketChannel
FactorialClientHanlder (upstream handler)
<uses>
ChannelSink
AbstractChannel
SocketChannelConfig
BlockingQueue
SimpleChannelUpstreamHandler
ChannelDownstreamHandler
ChannelHandler
Uses Channels class to write.

Client

```
ClientBootstrap: setPipelineFactory(FactorialClientPipelineFactory
DefaultChannelPipeline addLast(decoder, BigIntegerDecoder)
DefaultChannelPipeline addLast(encoder, NumberEncoder)
DefaultChannelPipeline addLast(handler, FactorialClientHandler)
Jul 19, 2009 12:22:40 PM org.jboss.netty.example.factorial.FactorialClientHandler handleUpstream
INFO: [id: 0x00e2cb55] OPEN
Jul 19, 2009 12:22:40 PM org.jboss.netty.example.factorial.FactorialClientHandler handleUpstream
INFO: [id: 0x00e2cb55, /127.0.0.1:49605 => localhost/127.0.0.1:8080] BOUND: /127.0.0.1:49605
Jul 19, 2009 12:22:40 PM org.jboss.netty.example.factorial.FactorialClientHandler handleUpstream
INFO: [id: 0x00e2cb55, /127.0.0.1:49605 => localhost/127.0.0.1:8080] CONNECTED: localhost/127.0.0.1:8080
CHANNEL CONNECTED: FactorialClientHandler channelConnected(DefaultChannelHandlerContext, UpstreamChannelStateEvent)
-------------FactorialClientHandler: sendNumber(1)
-------------NumberEncoder encode(DefaultChannelHandlerContext, NioClientSocketChannel)
-------------FactorialClientHandler: sendNumber(2)
-------------NumberEncoder encode(DefaultChannelHandlerContext, NioClientSocketChannel)
-------------FactorialClientHandler: sendNumber(3)
-------------NumberEncoder encode(DefaultChannelHandlerContext, NioClientSocketChannel)
-------------FactorialClientHandler: sendNumber(4)
-------------NumberEncoder encode(DefaultChannelHandlerContext, NioClientSocketChannel)
-------------FactorialClientHandler: sendNumber(5)
-------------NumberEncoder encode(DefaultChannelHandlerContext, NioClientSocketChannel)
-------------BigIntegerDecoder decode(NioClientSocketChannel, BigEndianHeapChannelBuffer): 1
-------------FactorialClientHandler messageReceived(DefaultChannelHandlerContext, UpstreamMessageEvent): message received: 1
-------------BigIntegerDecoder decode(NioClientSocketChannel, BigEndianHeapChannelBuffer): 2
-------------FactorialClientHandler messageReceived(DefaultChannelHandlerContext, UpstreamMessageEvent): message received: 2
-------------BigIntegerDecoder decode(NioClientSocketChannel, BigEndianHeapChannelBuffer): 6
-------------FactorialClientHandler messageReceived(DefaultChannelHandlerContext, UpstreamMessageEvent): message received: 3
-------------BigIntegerDecoder decode(NioClientSocketChannel, BigEndianHeapChannelBuffer): 24
-------------FactorialClientHandler messageReceived(DefaultChannelHandlerContext, UpstreamMessageEvent): message received: 4
-------------BigIntegerDecoder decode(NioClientSocketChannel, BigEndianHeapChannelBuffer): 120
Jul 19, 2009 12:22:40 PM org.jboss.netty.example.factorial.FactorialClientHandler handleUpstream
INFO: [id: 0x00e2cb55] DISCONNECTED
Jul 19, 2009 12:22:40 PM org.jboss.netty.example.factorial.FactorialClientHandler handleUpstream
INFO: [id: 0x00e2cb55] UNBOUND
Jul 19, 2009 12:22:40 PM org.jboss.netty.example.factorial.FactorialClientHandler handleUpstream
INFO: [id: 0x00e2cb55] CLOSED
FactorialClient: Factorial of 5 is: 120 Future Aanswer: 120
```



Server

```
ServerBootstrap: setPipelineFactory(FactorialServerPipelineFactory
ServerBootstrap bind(InetSocketAddress)
DefaultChannelPipeline addLast(binder, Binder)
DefaultChannelPipeline addLast(decoder, BigIntegerDecoder)
DefaultChannelPipeline addLast(encoder, NumberEncoder)
DefaultChannelPipeline addLast(handler, FactorialServerHandler)
Jul 19, 2009 12:22:40 PM org.jboss.netty.example.factorial.FactorialServerHandler handleUpstream
INFO: [id: 0x00baa466, /127.0.0.1:49605 => /127.0.0.1:8080] OPEN
Jul 19, 2009 12:22:40 PM org.jboss.netty.example.factorial.FactorialServerHandler handleUpstream
INFO: [id: 0x00baa466, /127.0.0.1:49605 => /127.0.0.1:8080] BOUND: /127.0.0.1:8080
Jul 19, 2009 12:22:40 PM org.jboss.netty.example.factorial.FactorialServerHandler handleUpstream
INFO: [id: 0x00baa466, /127.0.0.1:49605 => /127.0.0.1:8080] CONNECTED: /127.0.0.1:49605
-------------BigIntegerDecoder decode(NioAcceptedSocketChannel, BigEndianHeapChannelBuffer): 1
A-------------FactorialServerHandler messageReceived(DefaultChannelHandlerContext, UpstreamMessageEvent)
B-------------FactorialServerHandler last multiplier 1
C-------------FactorialServerHandler messageReceived WRITE 1
-------------NumberEncoder encode(DefaultChannelHandlerContext, NioAcceptedSocketChannel)
-------------BigIntegerDecoder decode(NioAcceptedSocketChannel, BigEndianHeapChannelBuffer): 2
A-------------FactorialServerHandler messageReceived(DefaultChannelHandlerContext, UpstreamMessageEvent)
B-------------FactorialServerHandler last multiplier 2
C-------------FactorialServerHandler messageReceived WRITE 2
-------------NumberEncoder encode(DefaultChannelHandlerContext, NioAcceptedSocketChannel)
-------------BigIntegerDecoder decode(NioAcceptedSocketChannel, BigEndianHeapChannelBuffer): 3
A-------------FactorialServerHandler messageReceived(DefaultChannelHandlerContext, UpstreamMessageEvent)
B-------------FactorialServerHandler last multiplier 3
C-------------FactorialServerHandler messageReceived WRITE 6
-------------NumberEncoder encode(DefaultChannelHandlerContext, NioAcceptedSocketChannel)
-------------BigIntegerDecoder decode(NioAcceptedSocketChannel, BigEndianHeapChannelBuffer): 4
A-------------FactorialServerHandler messageReceived(DefaultChannelHandlerContext, UpstreamMessageEvent)
B-------------FactorialServerHandler last multiplier 4
C-------------FactorialServerHandler messageReceived WRITE 24
-------------NumberEncoder encode(DefaultChannelHandlerContext, NioAcceptedSocketChannel)
-------------BigIntegerDecoder decode(NioAcceptedSocketChannel, BigEndianHeapChannelBuffer): 5
A-------------FactorialServerHandler messageReceived(DefaultChannelHandlerContext, UpstreamMessageEvent)
B-------------FactorialServerHandler last multiplier 5
C-------------FactorialServerHandler messageReceived WRITE 120
-------------NumberEncoder encode(DefaultChannelHandlerContext, NioAcceptedSocketChannel)
Jul 19, 2009 12:22:40 PM org.jboss.netty.example.factorial.FactorialServerHandler handleUpstream
INFO: [id: 0x00baa466] DISCONNECTED
Jul 19, 2009 12:22:40 PM org.jboss.netty.example.factorial.FactorialServerHandler channelDisconnected
INFO: On Channel disconnected: factorial of 5 is: 120
Jul 19, 2009 12:22:40 PM org.jboss.netty.example.factorial.FactorialServerHandler handleUpstream
INFO: [id: 0x00baa466] UNBOUND
Jul 19, 2009 12:22:40 PM org.jboss.netty.example.factorial.FactorialServerHandler handleUpstream
INFO: [id: 0x00baa466] CLOSED
```