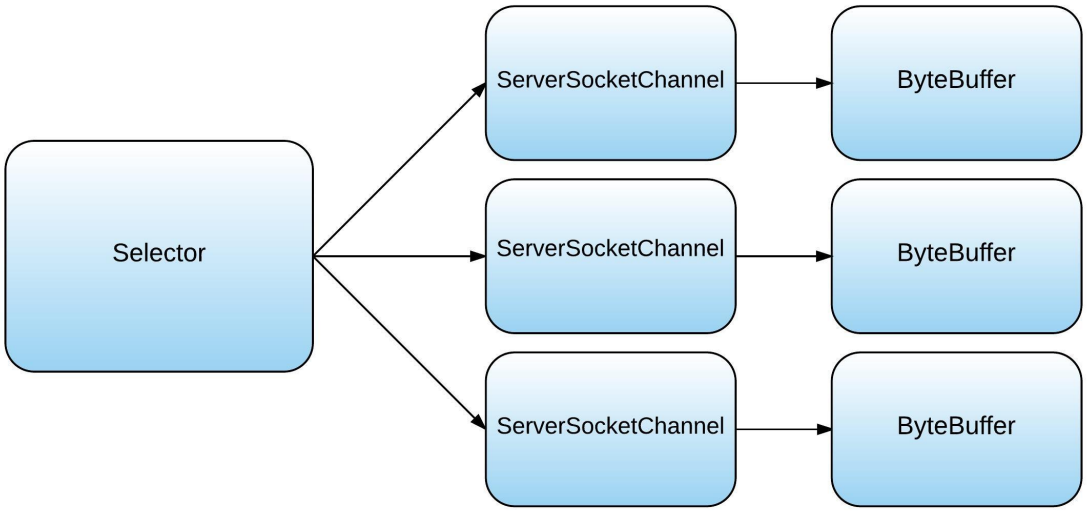
Java NIO

Java NIO

- NIO, is the New IO package created as part of Java 1.4.
- It is an improvement over stream-based IO because of its non-blocking nature.
- One thread can handle connections from multiple clients - no need to start a new thread for every new connection.
- However, more complex than stream-based IO.

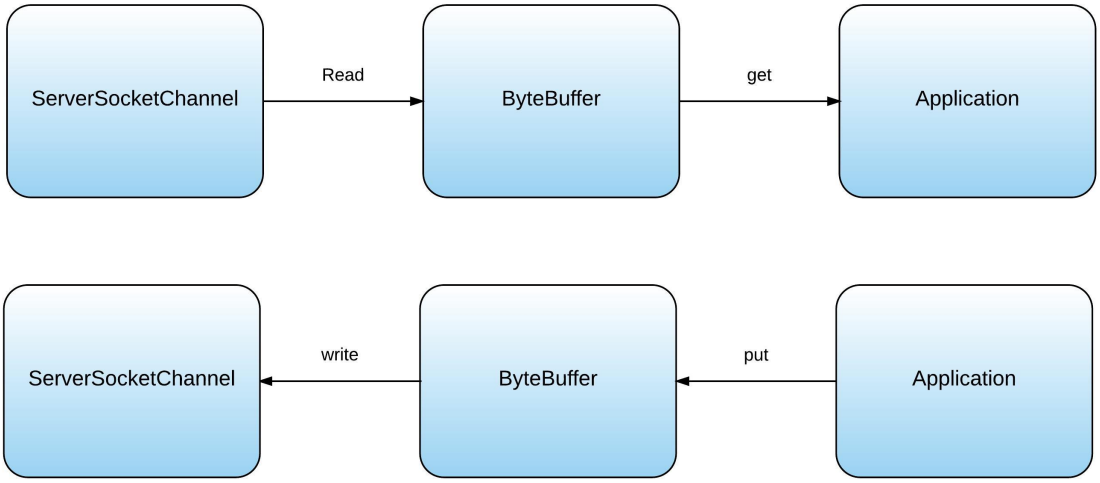
Overview

- NIO uses selectors, channels and buffers.
- Channels come in many flavors. We use ServerSocketChannel to implement a TCP socket server.
- Buffers also come in many flavors (ByteBuffer, CharBuffer, IntBuffer...). We use ByteBuffers.
- Selectors are event based: register interest in selectors and receive event (accept new conection, socket data received, socket data ready to be sent).



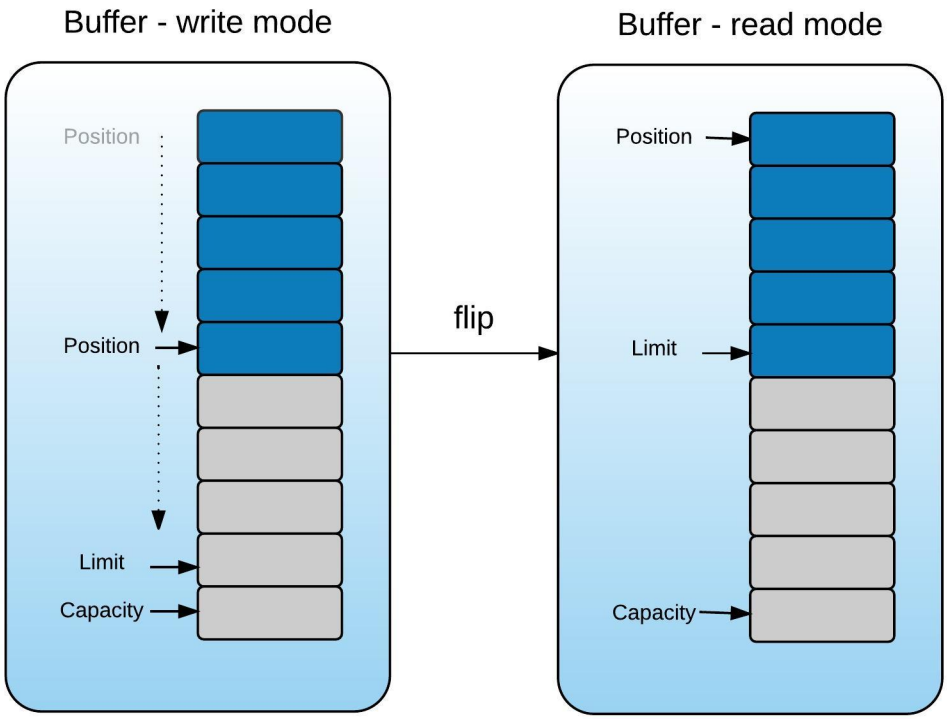
Channels and buffers

- Buffers always acts as intermediaries between application and the channel.
- Data is read from channels to buffers.
- Data is written from buffers to channels.



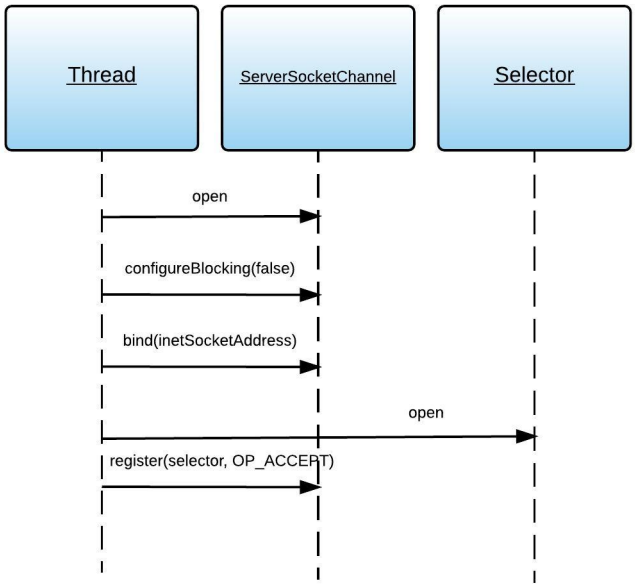
Buffer IO

- Buffers have read and write modes.
- Space is allocated in the buffer (denoted by capacity).
- As data is written to the buffer, the position moves down. The limit is the capacity.
- Once data has been written, it can be read by either the application or the channel.
- Flip is invoked to change its mode. Position resets to the start and limit is set to the last position representing the last byte of data written.
- Attempting to write passed the limit results in a BufferOverflowException.
- Attempting to read in write mode results in BufferUnderflowException.



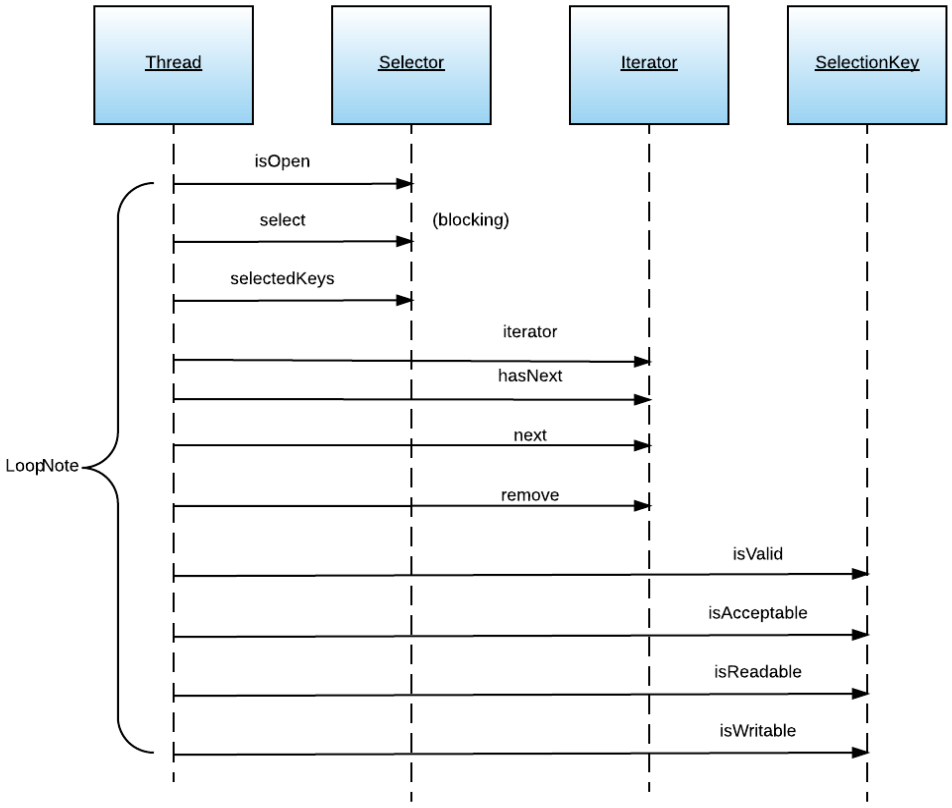
Binding a channel to a port

- Channels are bound to selectors.



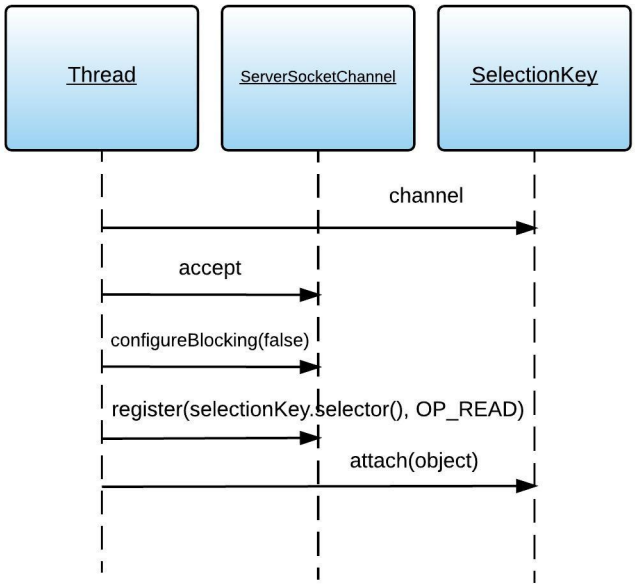
Listening for events

- One thread can listens to events.



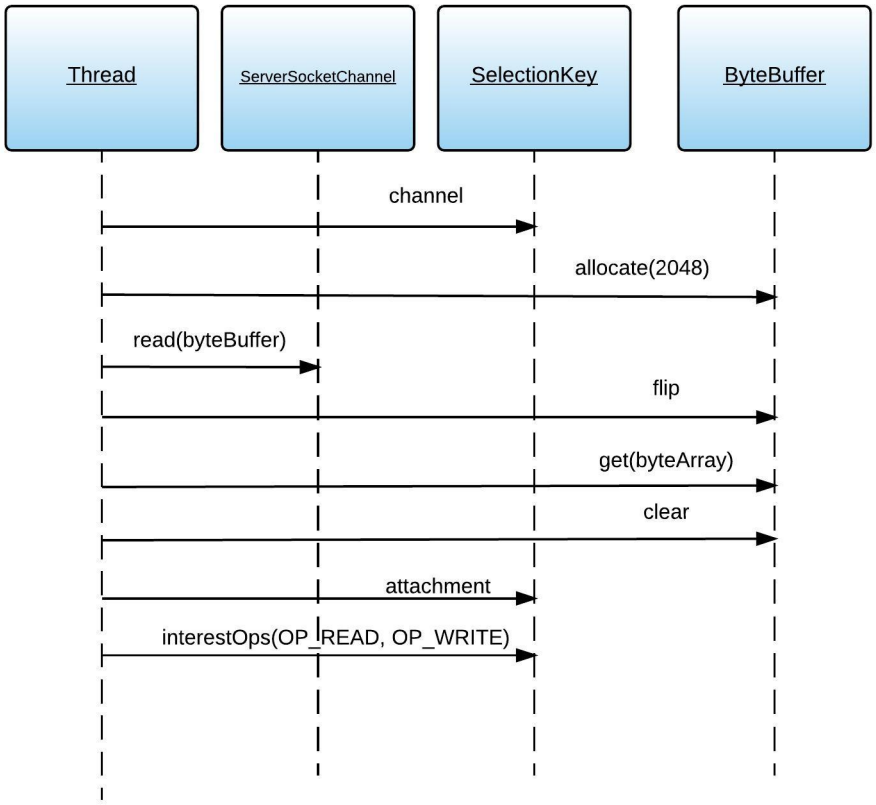
Establishing new connections

- Register to receive read events for new connection.
- Connection data can be attached to the selection key and retrieved later on.



Read data from connections

- Read from channel into buffer.
- Register interest in writing to respond to client.



Writing data to connections

- Write into channel from buffer.
- Remove interest in write events.