Networking in Java with NIO and Netty

KANSTANTSIN SLISENKA
SENIOR SOFTWARE ENGINEER

DECEMBER 22, 2016

About me

Kanstantsin Slisenka

- Java backend developer
- Speaker at Java tech talks

I AM INTERESTED IN

- Complex Java backend, SOA, databases
- High load, fault-tolerant, distributed systems

skype: kslisenko

kslisenko@gmail.com, kanstantsin_slisenka@epam.com

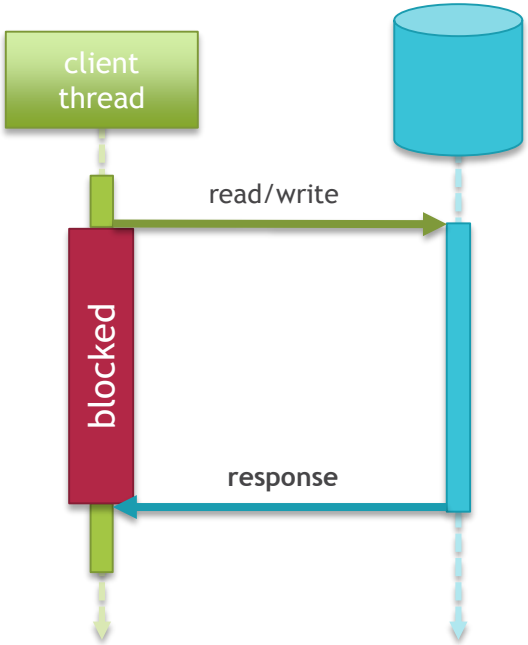
Agenda

1. Blocking vs non-blocking resources
2. Blocking IO: sockets
3. Non-blocking IO: NIO socket channels
4. Netty: high performance networking framework

Blocking vs non-blocking resources

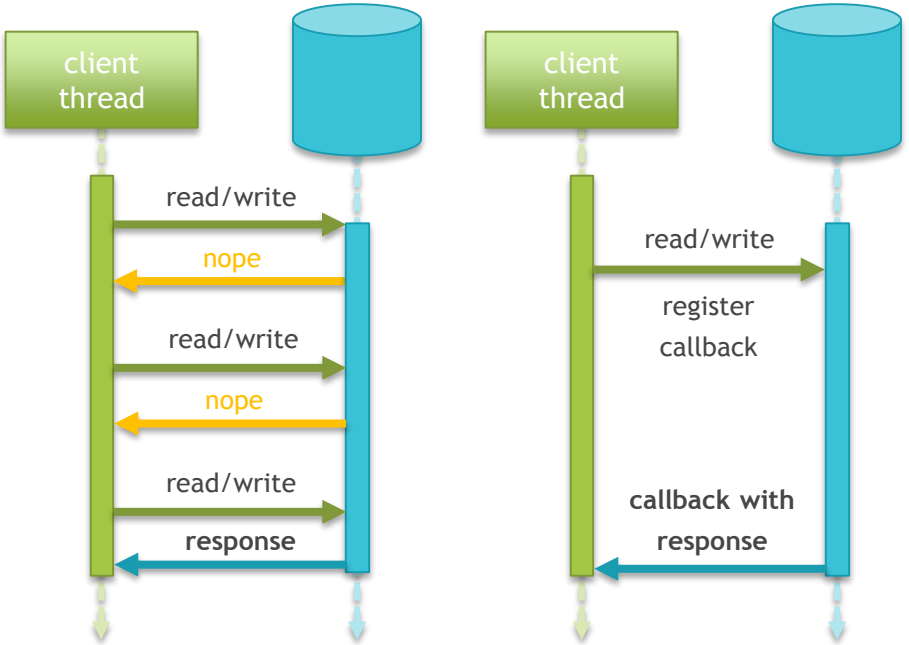
Blocking resources

Database transactions, RPC/WS calls

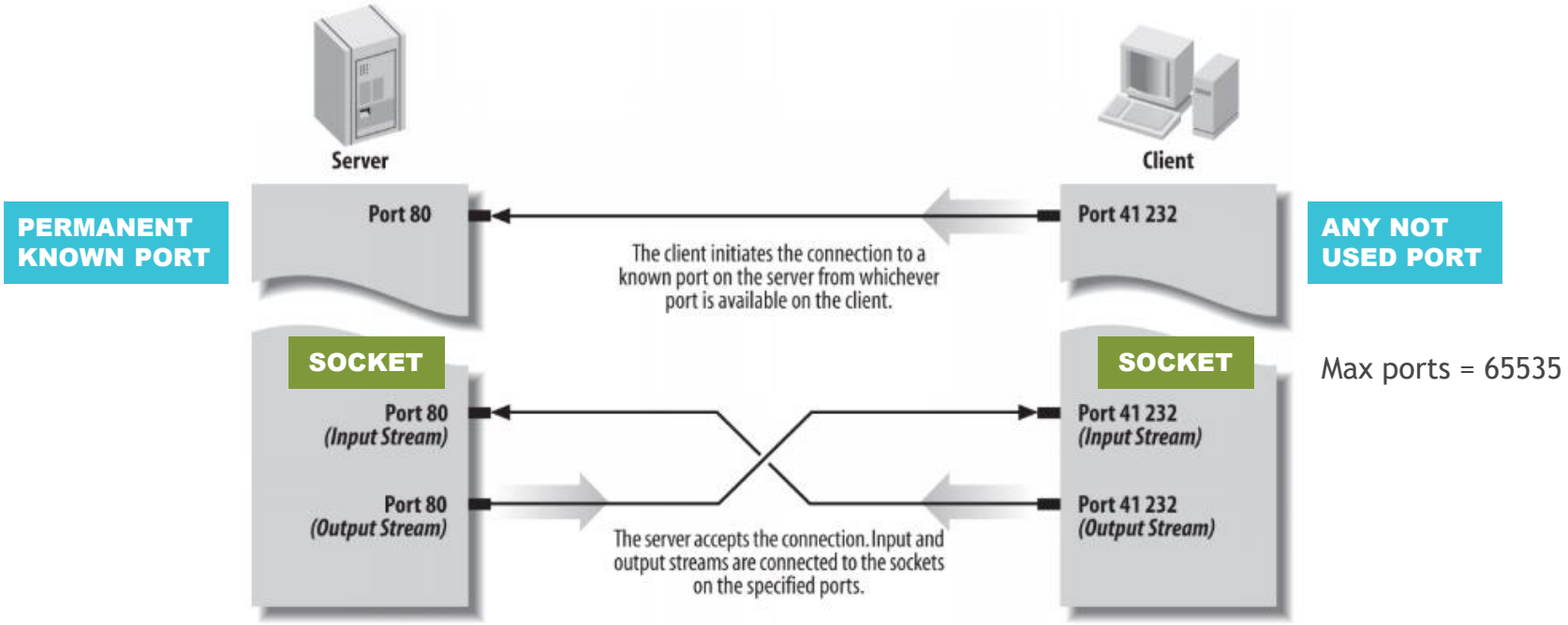


Non-blocking resources

Queues, WebSockets, Non-blocking data storage



Network sockets



Max ports = 65535

Sockets are peer-to-peer communication objects at both sides

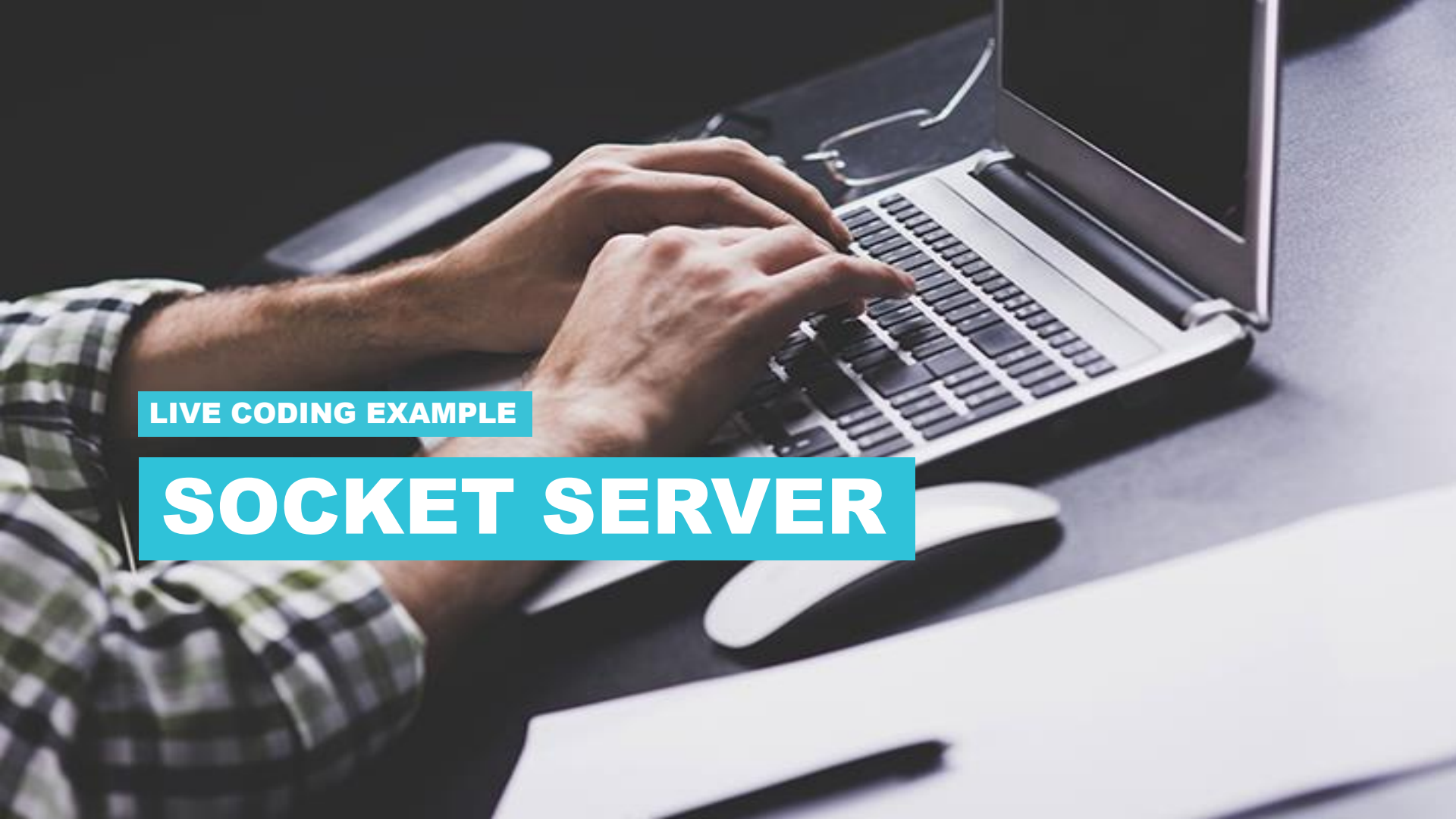
Java sockets use blocking IO

Client

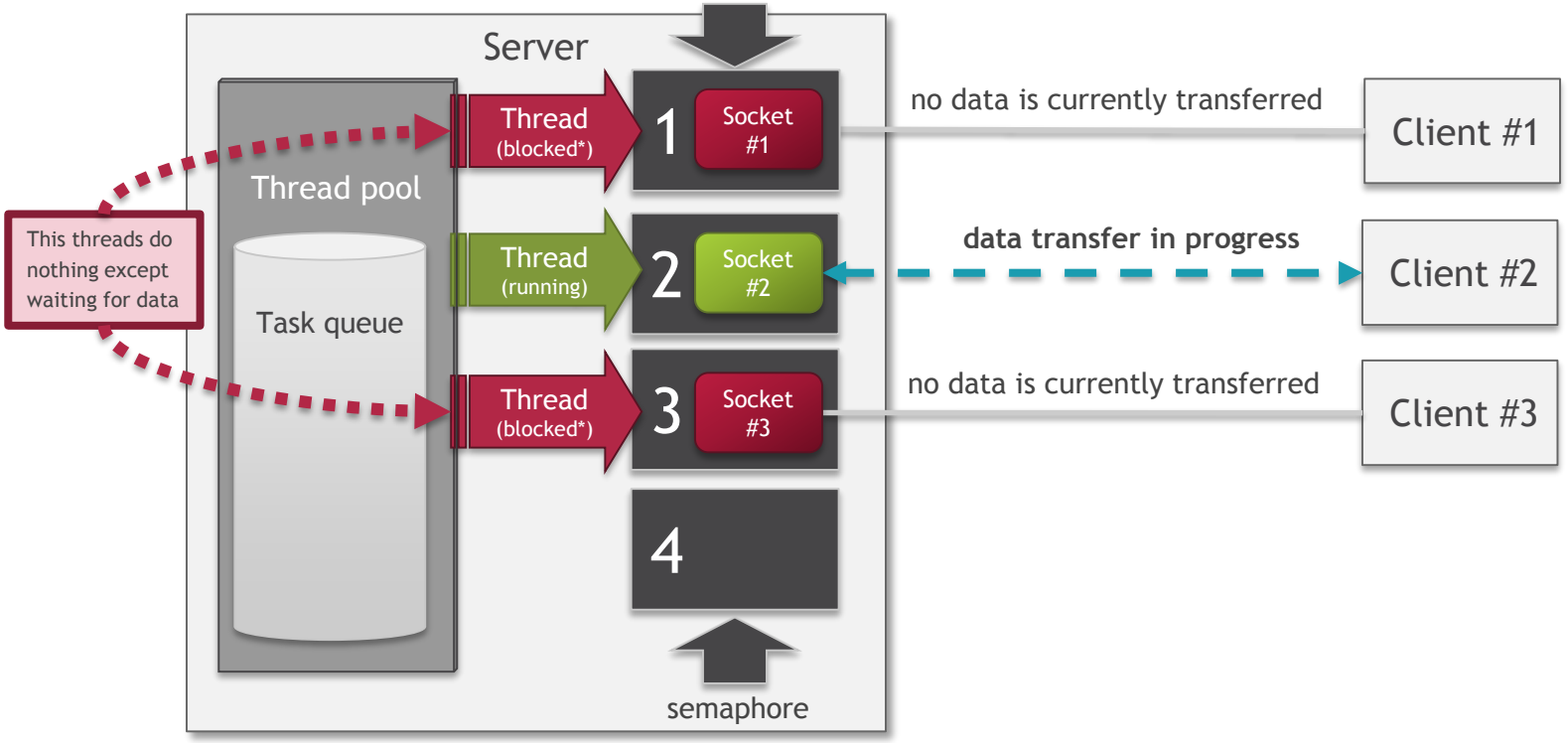
```
Socket socket = new Socket("localhost", 8080);
OutputStream out = socket.getOutputStream();
InputStream in = socket.getInputStream();

out.write("hello".getBytes()); // Blocking call

byte[] data = new byte[1024];
while (in.read(data) != -1) { // Blocking call
    handle(data);
}
```

Server

```
ServerSocket serverSocket = new ServerSocket(8080);
Socket socket = serverSocket.accept();
InputStream in = socket.getInputStream();
OutputStream out = socket.getOutputStream();

byte[] data = new byte[1024];
while (in.read(data) != -1) { // Blocking call
    out.write(data); // Blocking call
}
```

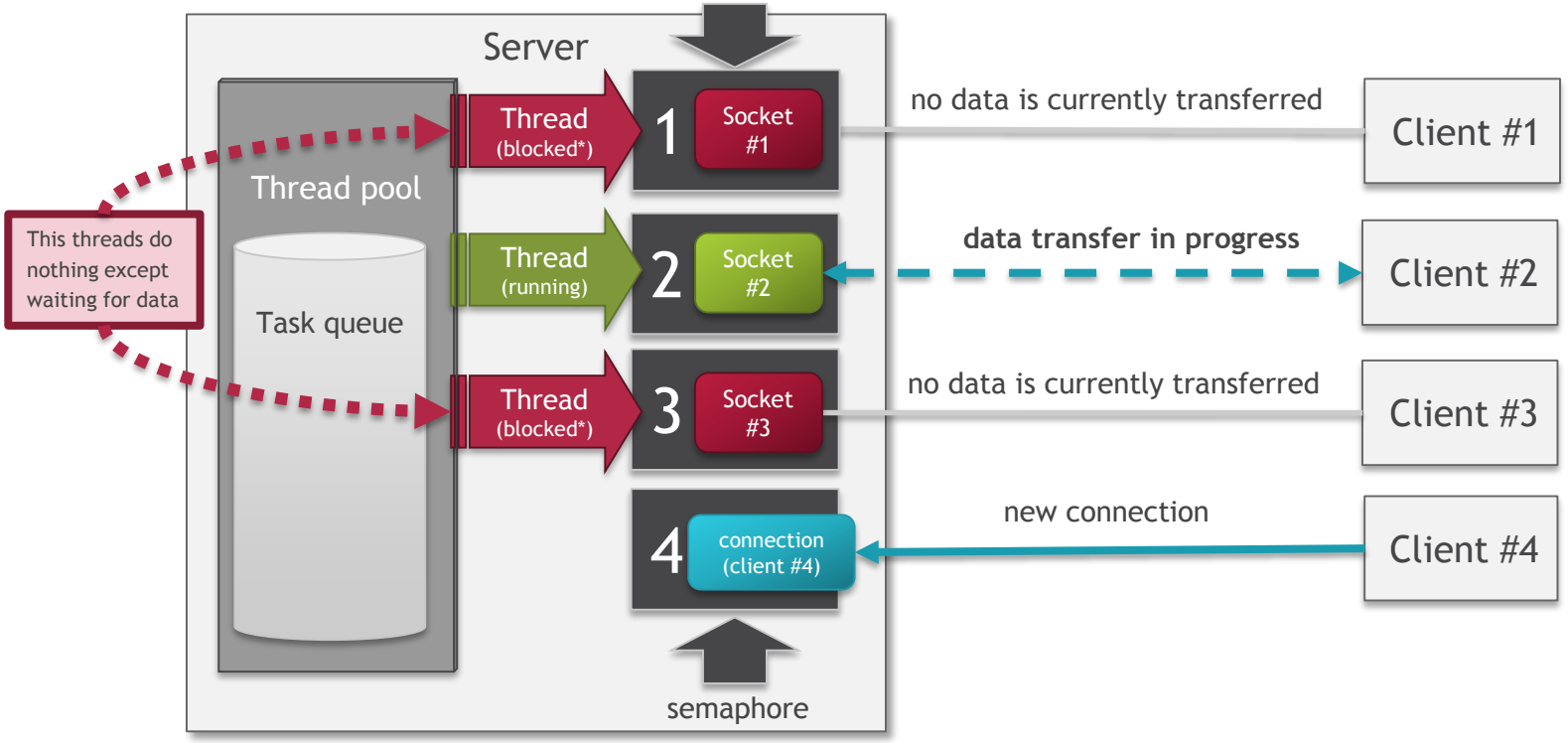

LIVE CODING EXAMPLE
SOCKET SERVER

Socket server architecture



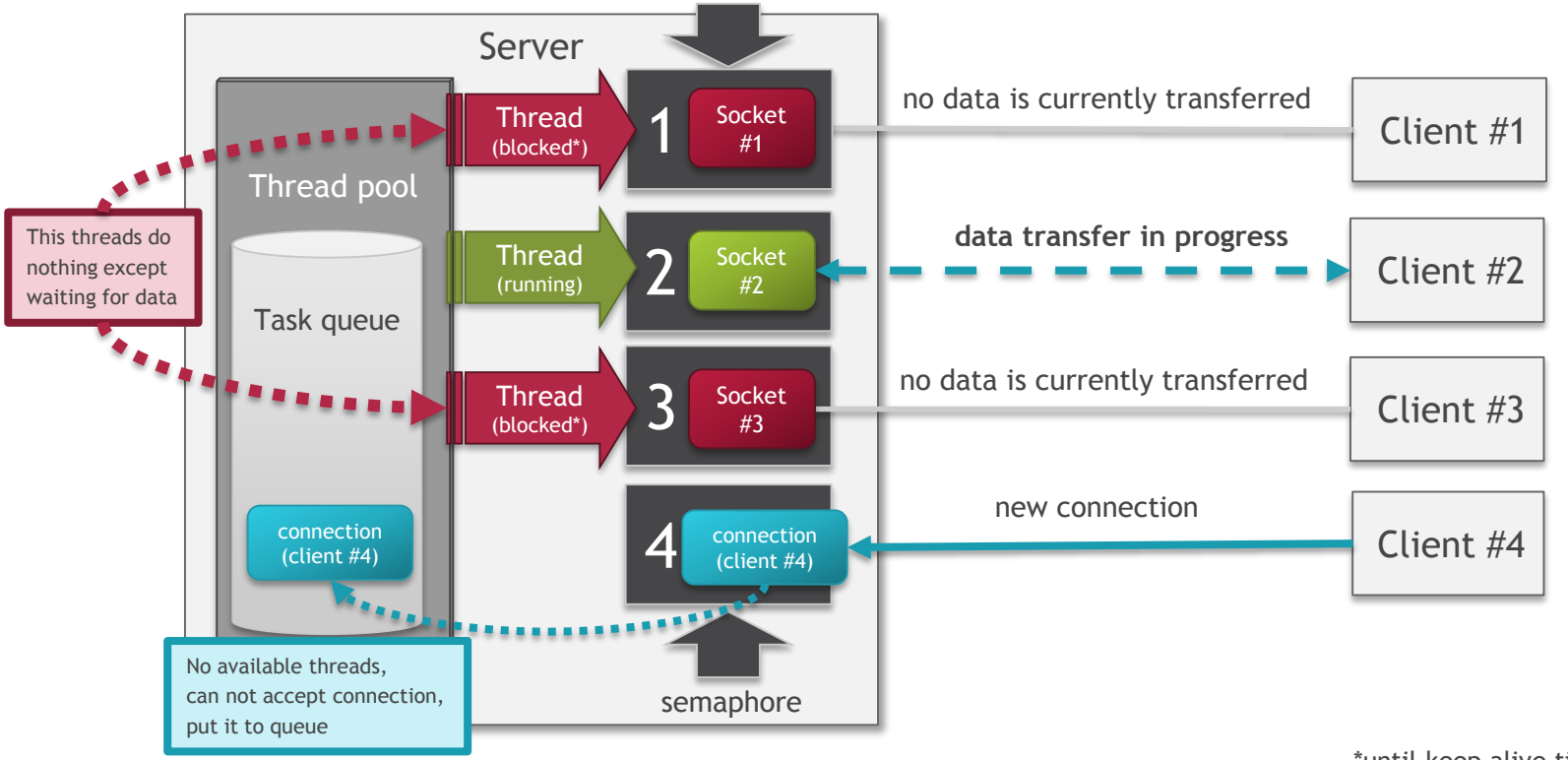
*until keep alive timeout

Socket server architecture



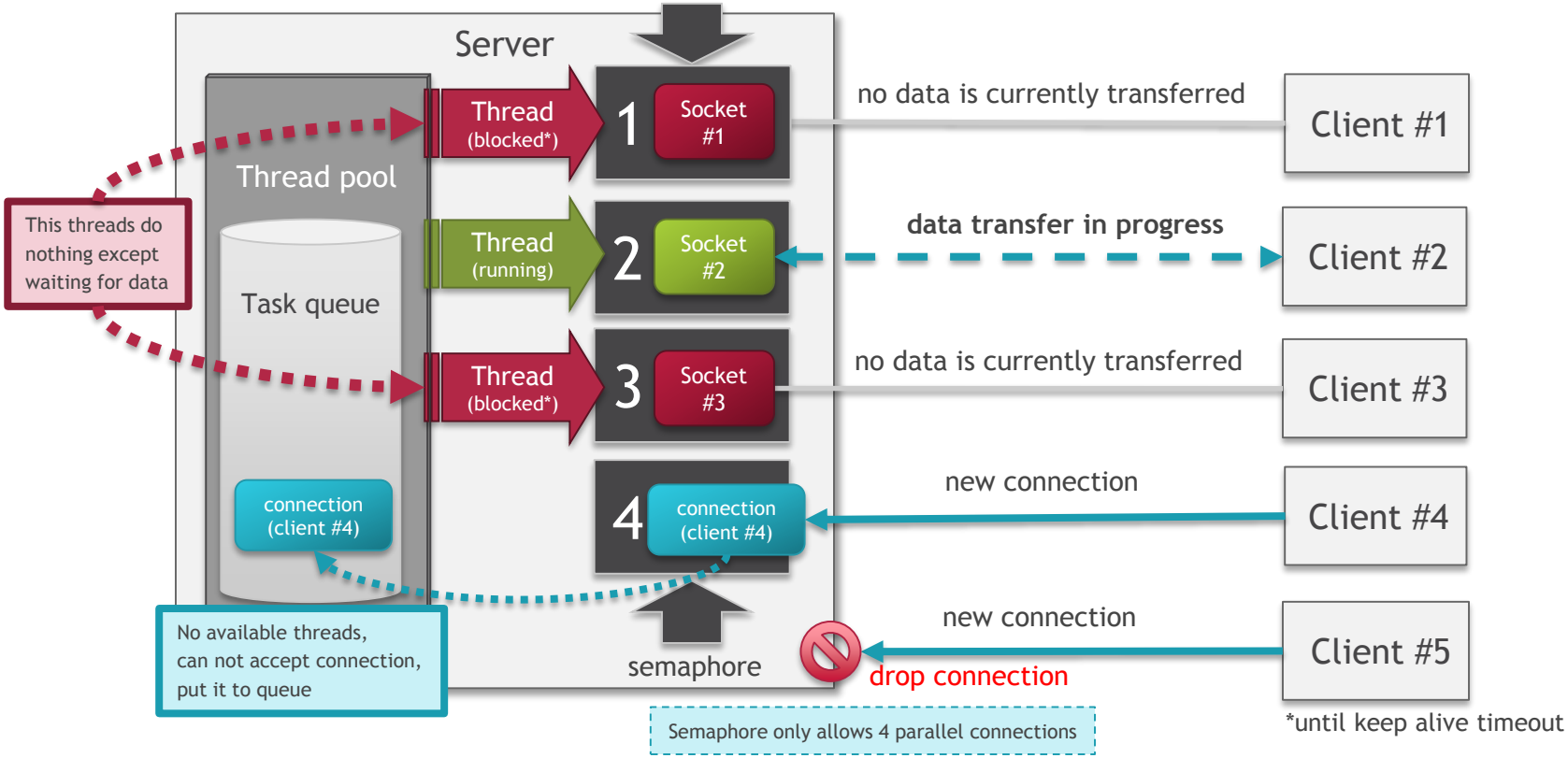
*until keep alive timeout

Socket server architecture



*until keep alive timeout

Socket server architecture



Socket server conclusion

Advantages

- **Simple development**

  because we are using well known Input/OutputStream API

- **No check for partially received messages**

  thread is blocked until fully reads message

Drawbacks

- **Too many threads**

  extra memory

  context switching overhead

- **Responsiveness**

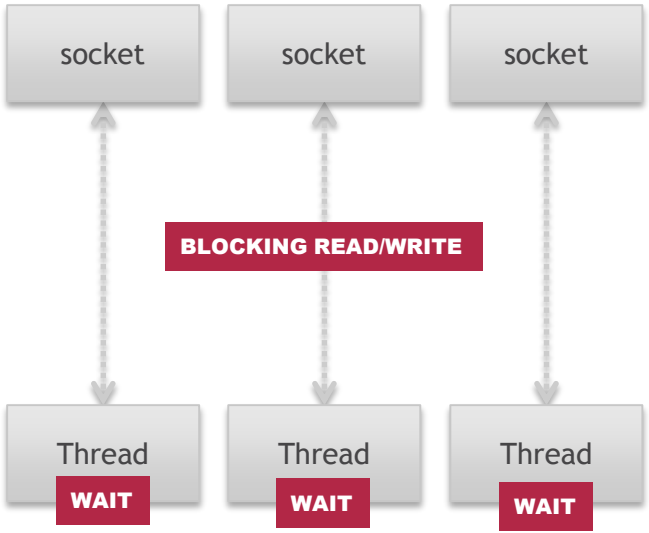
  when all threads are blocked putting new connections to the thread pool queue makes server unresponsive

- **Connections count limit**

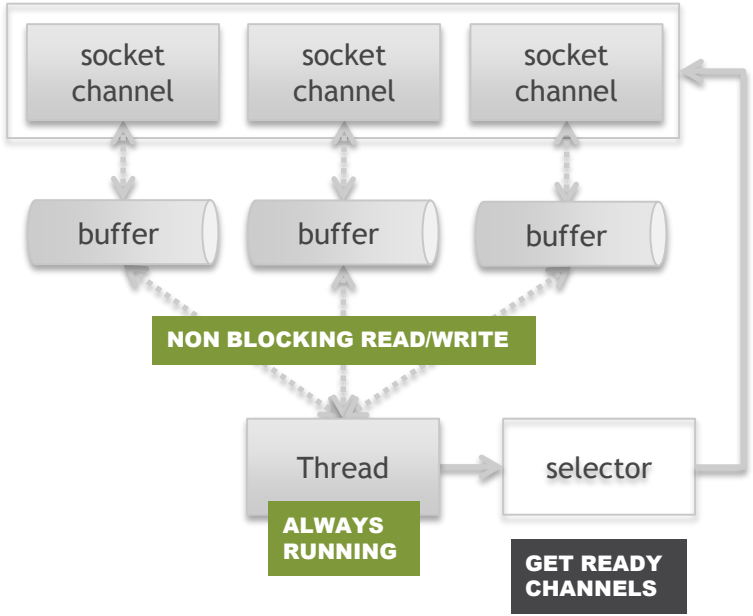
  semaphore prevents from over queuing, but limits number of parallel connections

IO vs NIO

IO (blocking)



NIO (non-blocking)



NIO byte buffer

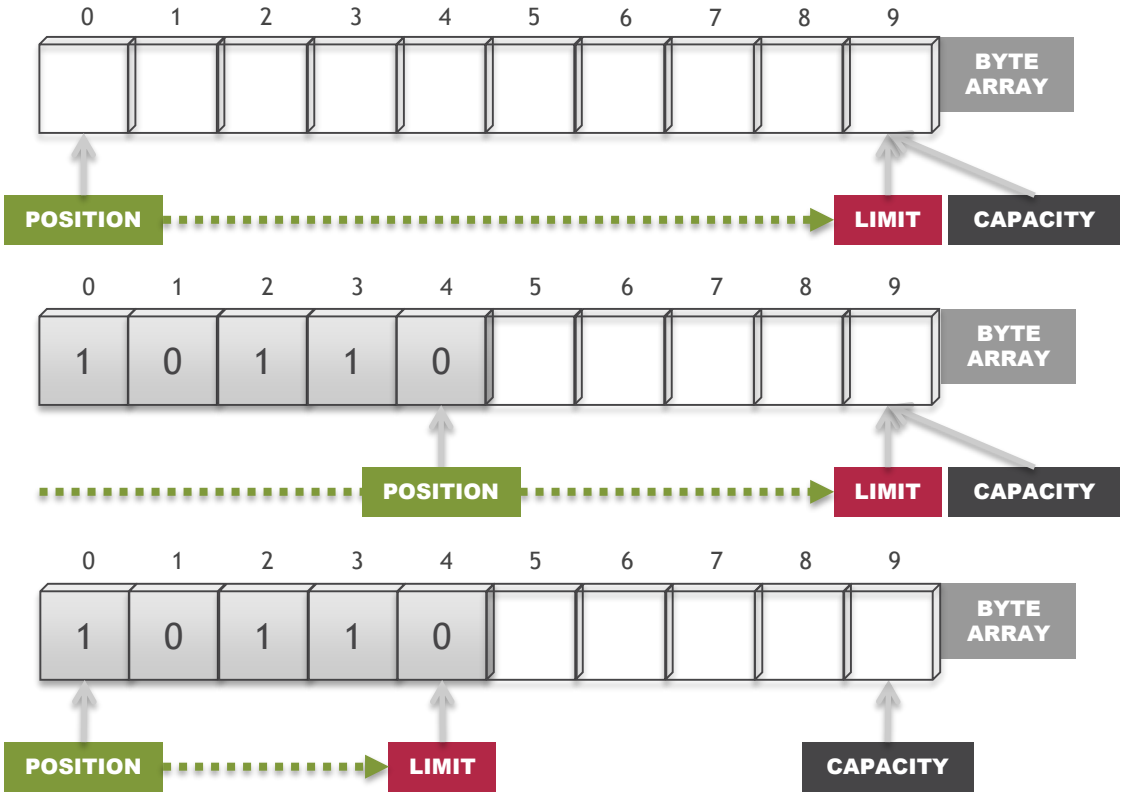
Empty buffer

Writing mode

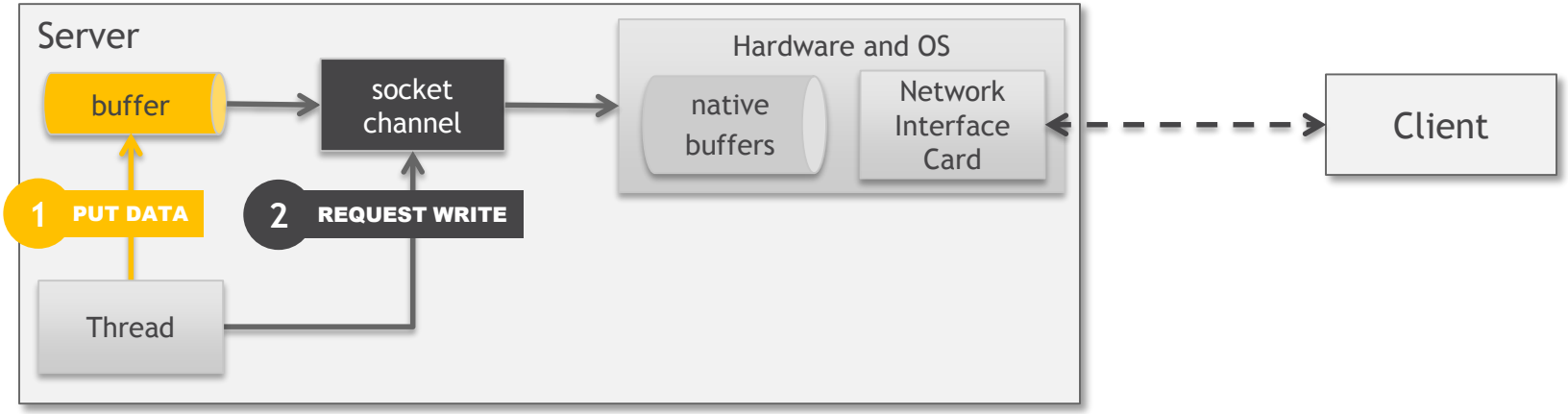
buffer.clear()

buffer.flip()

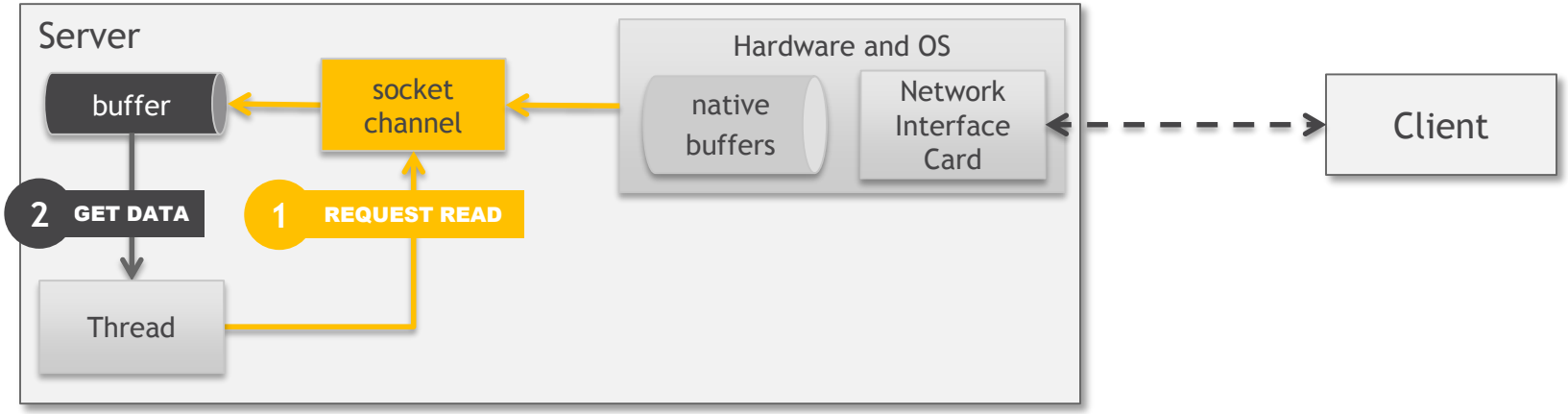
Reading mode



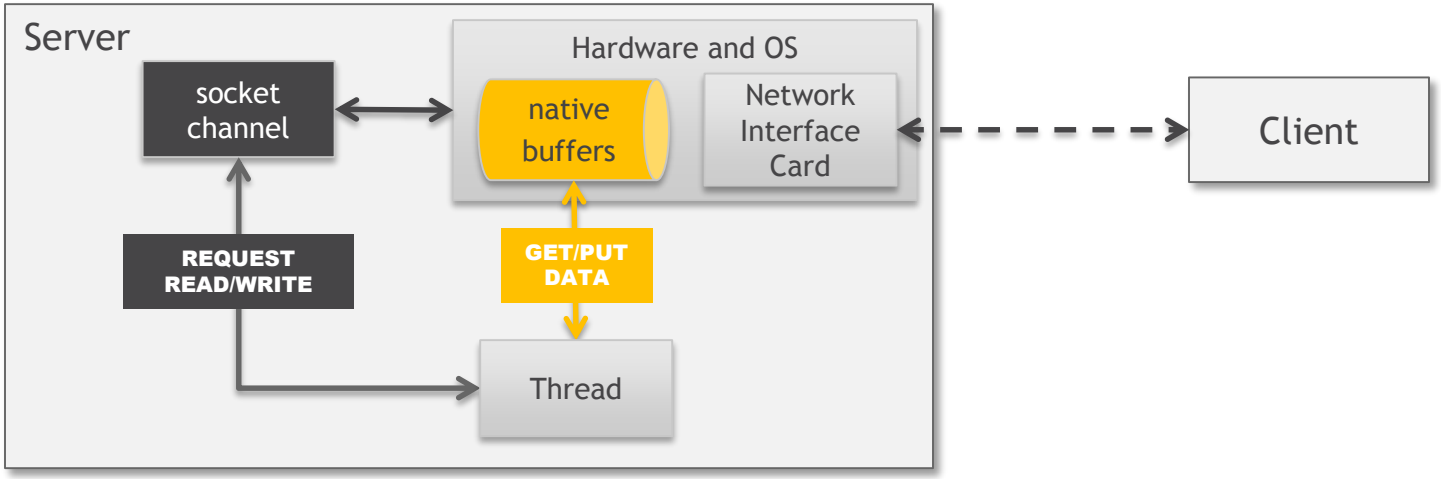
NIO socket channel: data sending



NIO socket channel: data receiving



NIO direct buffer

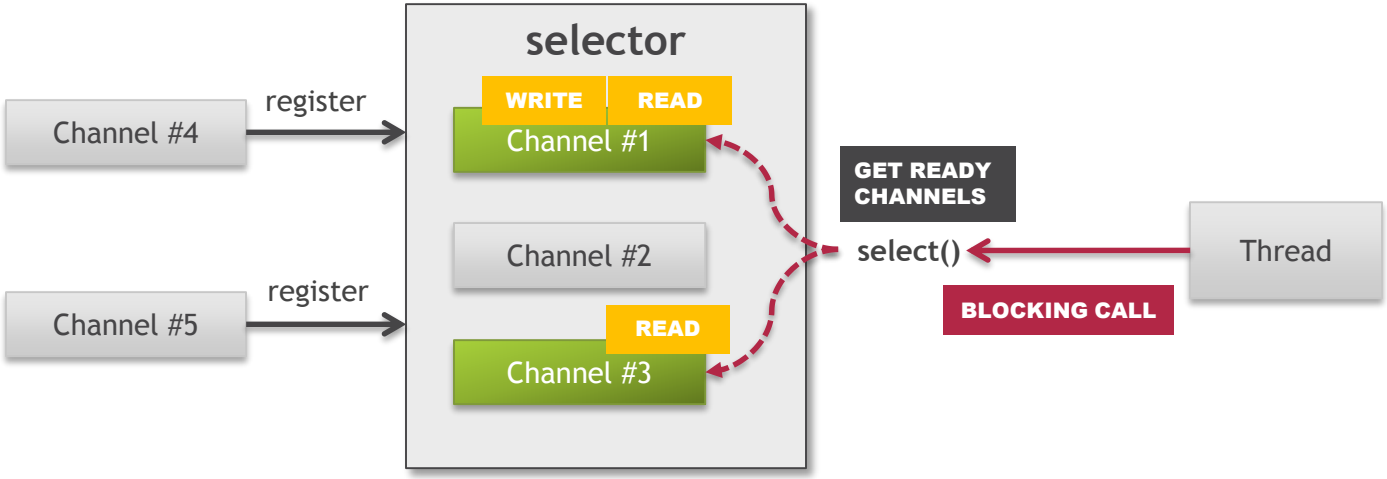


Lives in native memory

Cheep to read/write

Expensive to allocate/deallocate

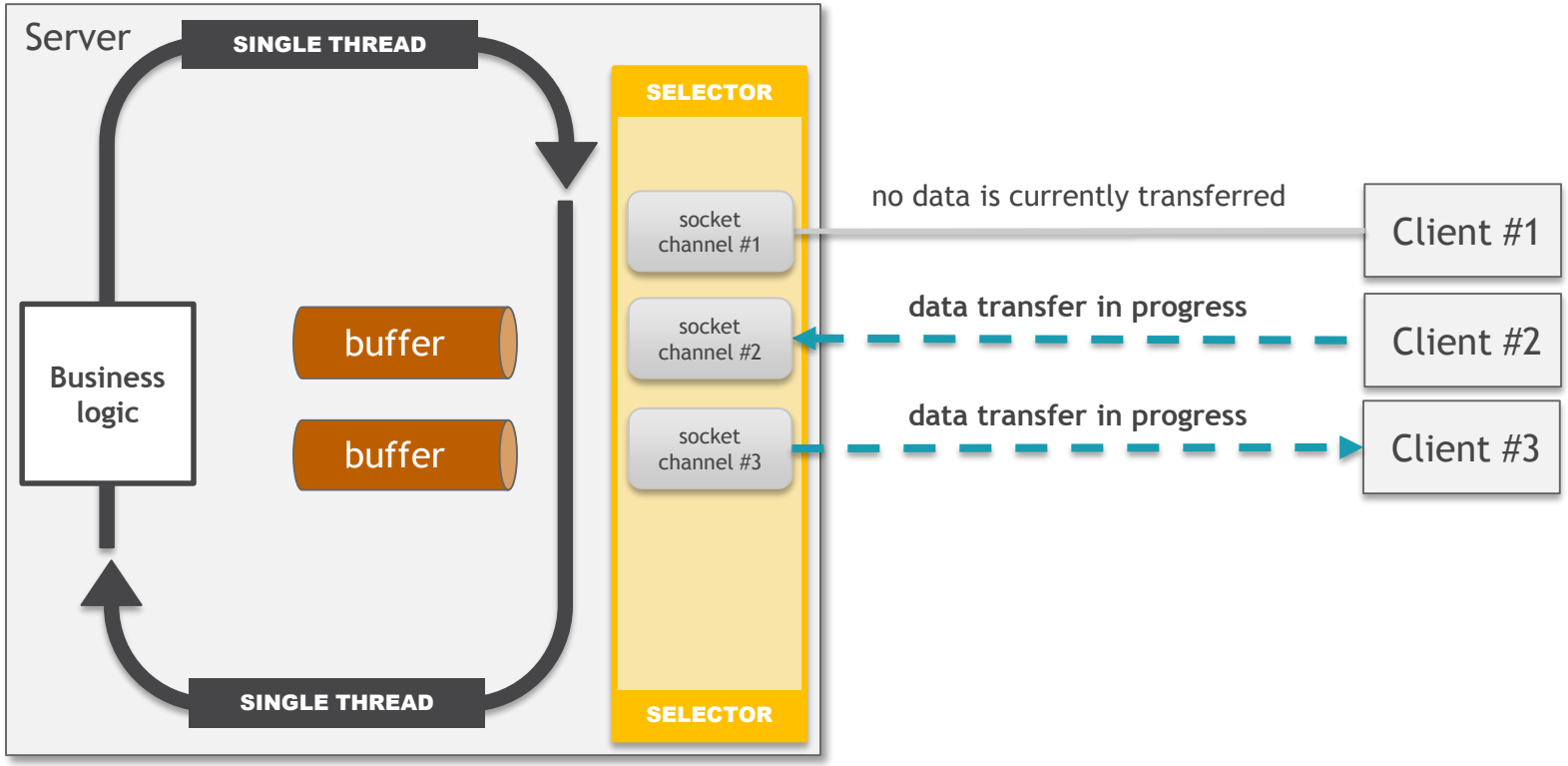
NIO selector



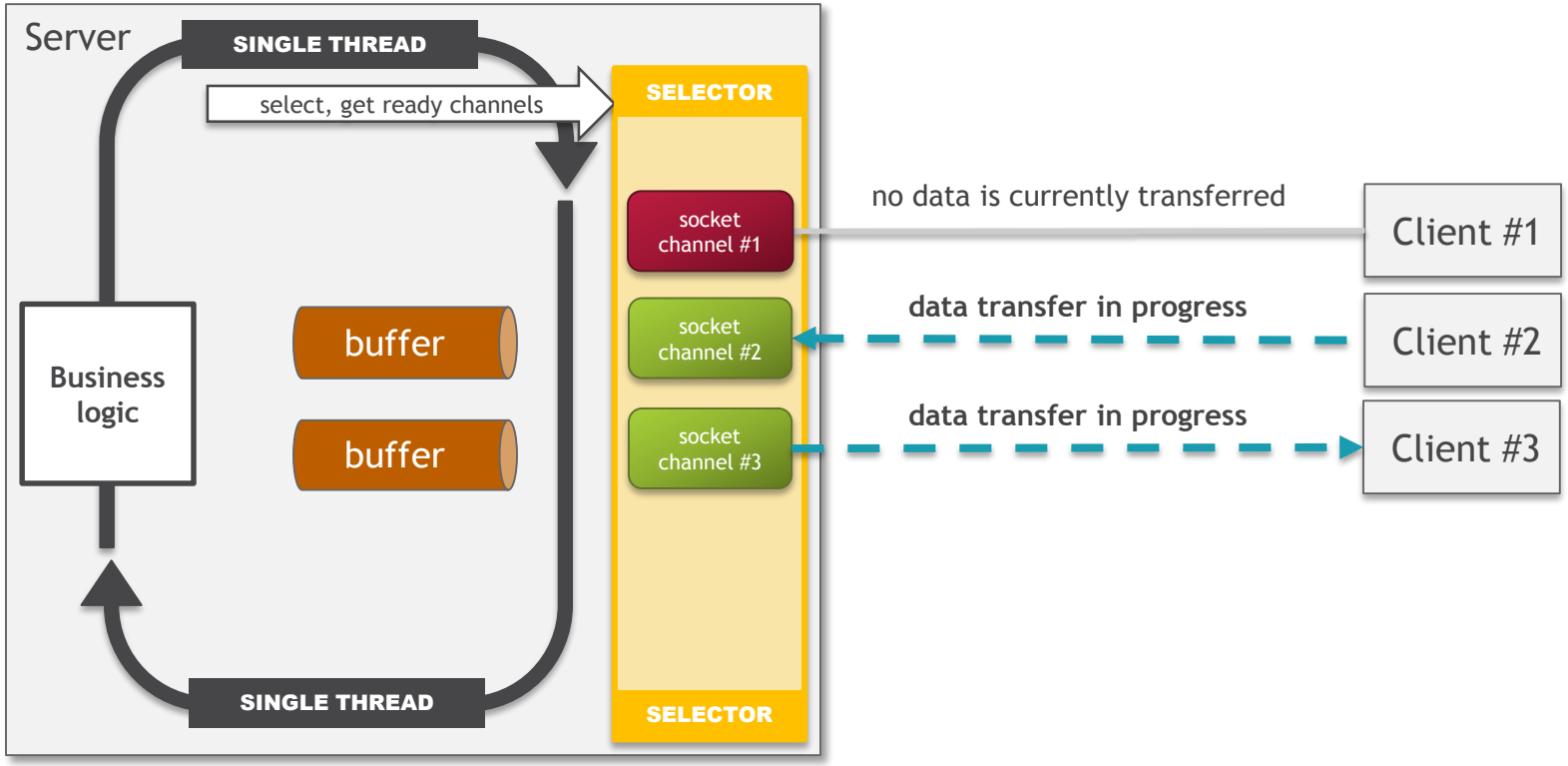
Selector blocks until something happens at least at one channel and returns ready channels


LIVE CODING EXAMPLE
NIO SERVER

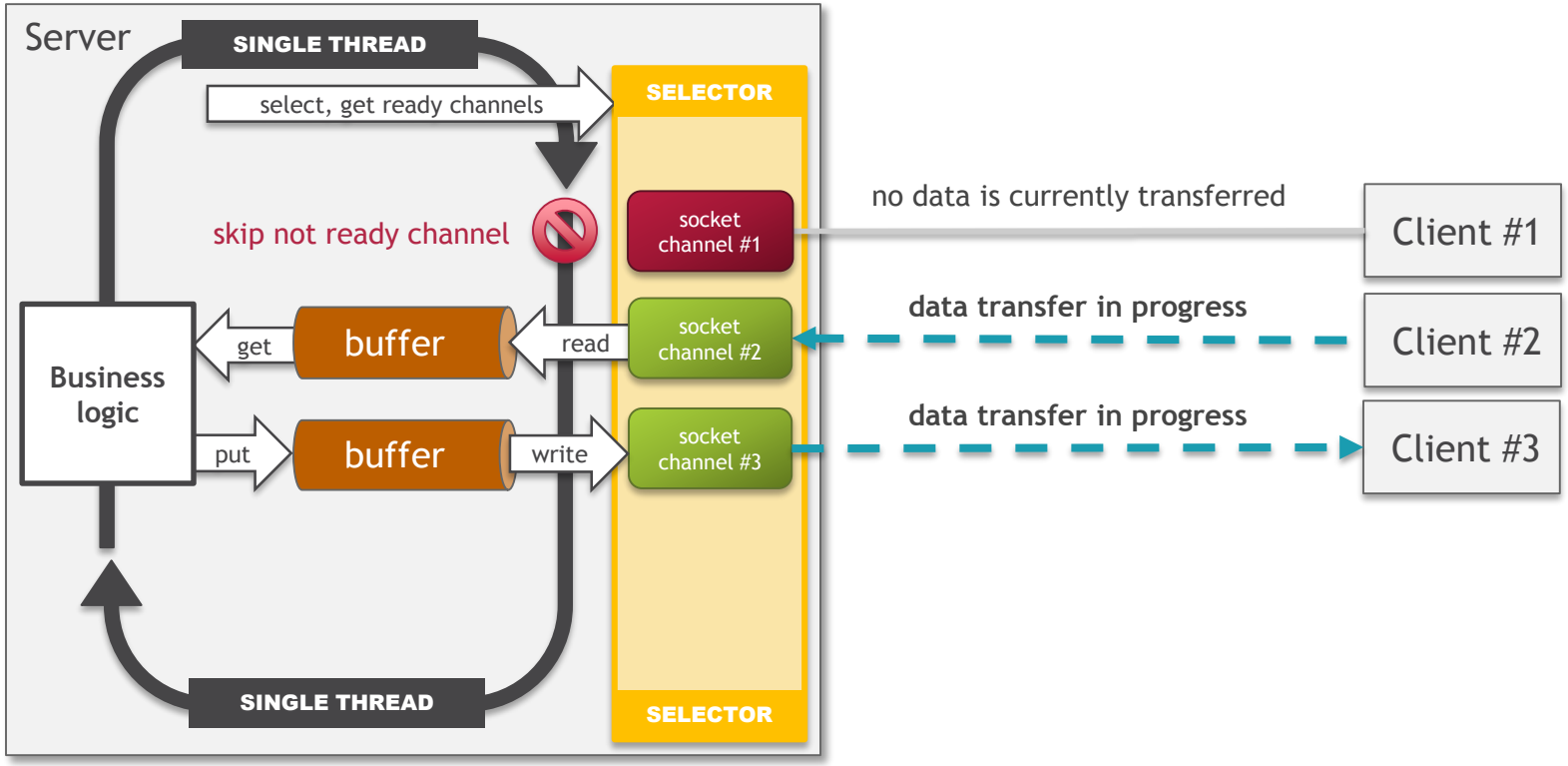
Basic NIO server architecture



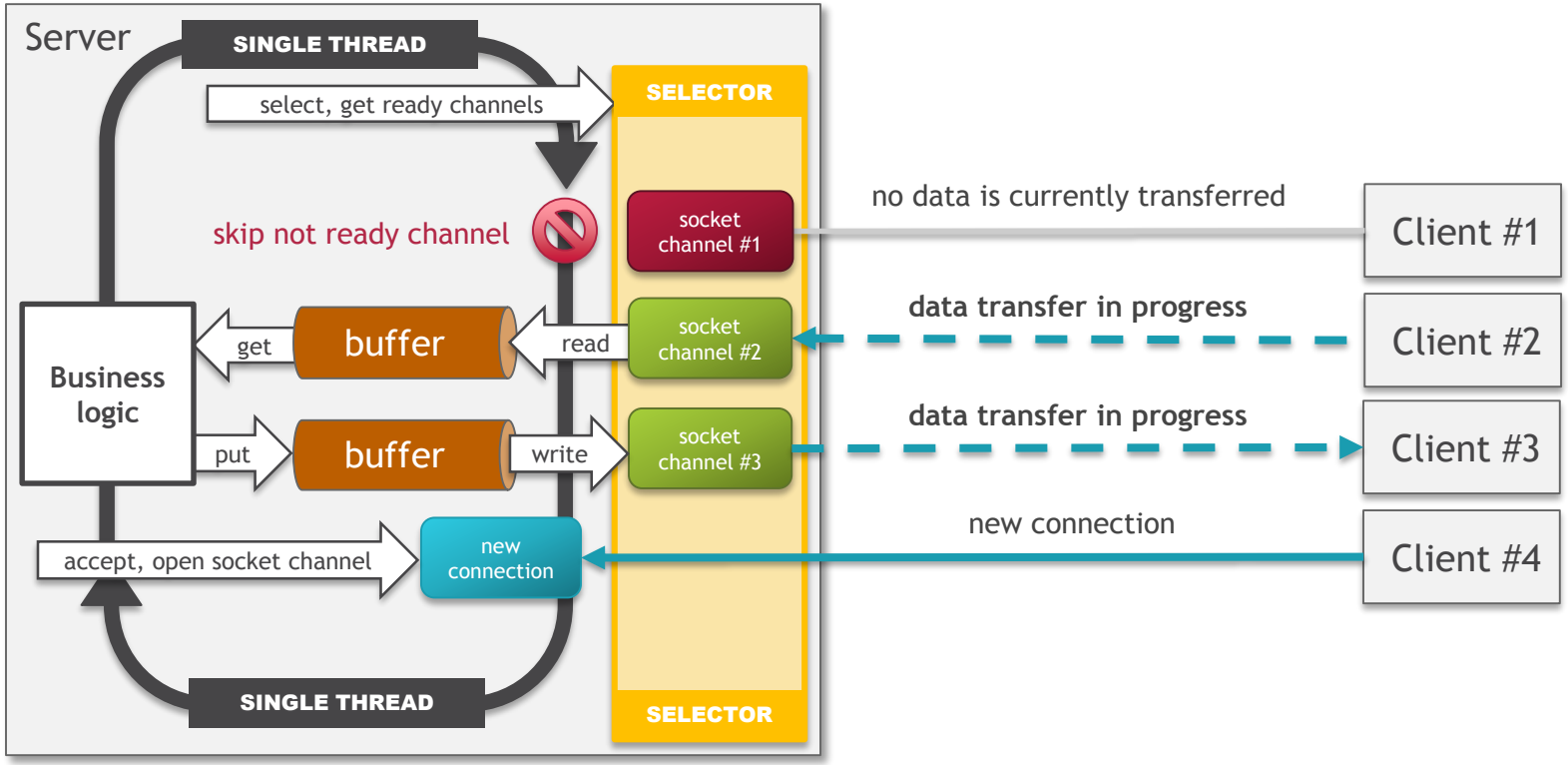
Basic NIO server architecture



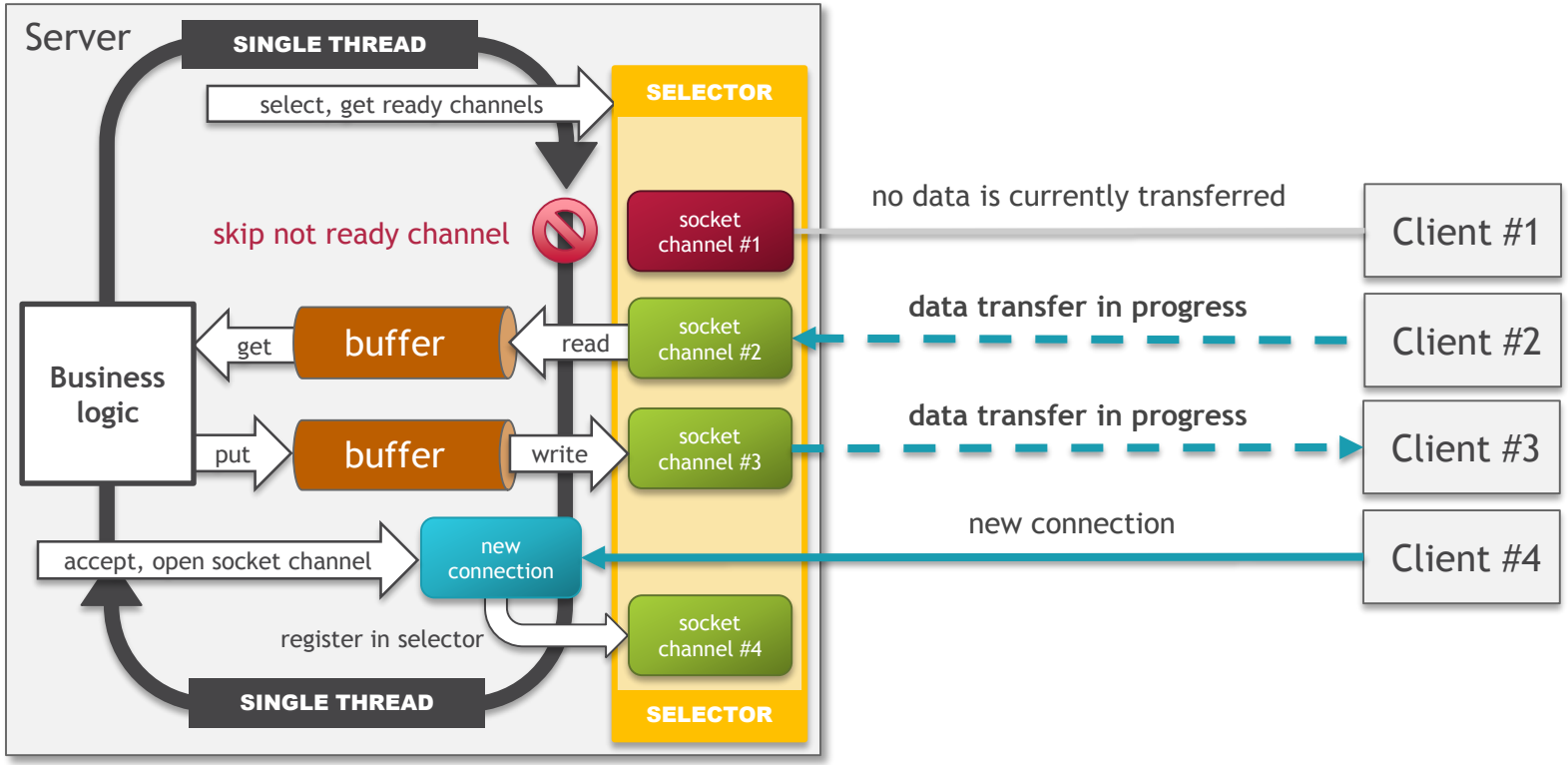
Basic NIO server architecture



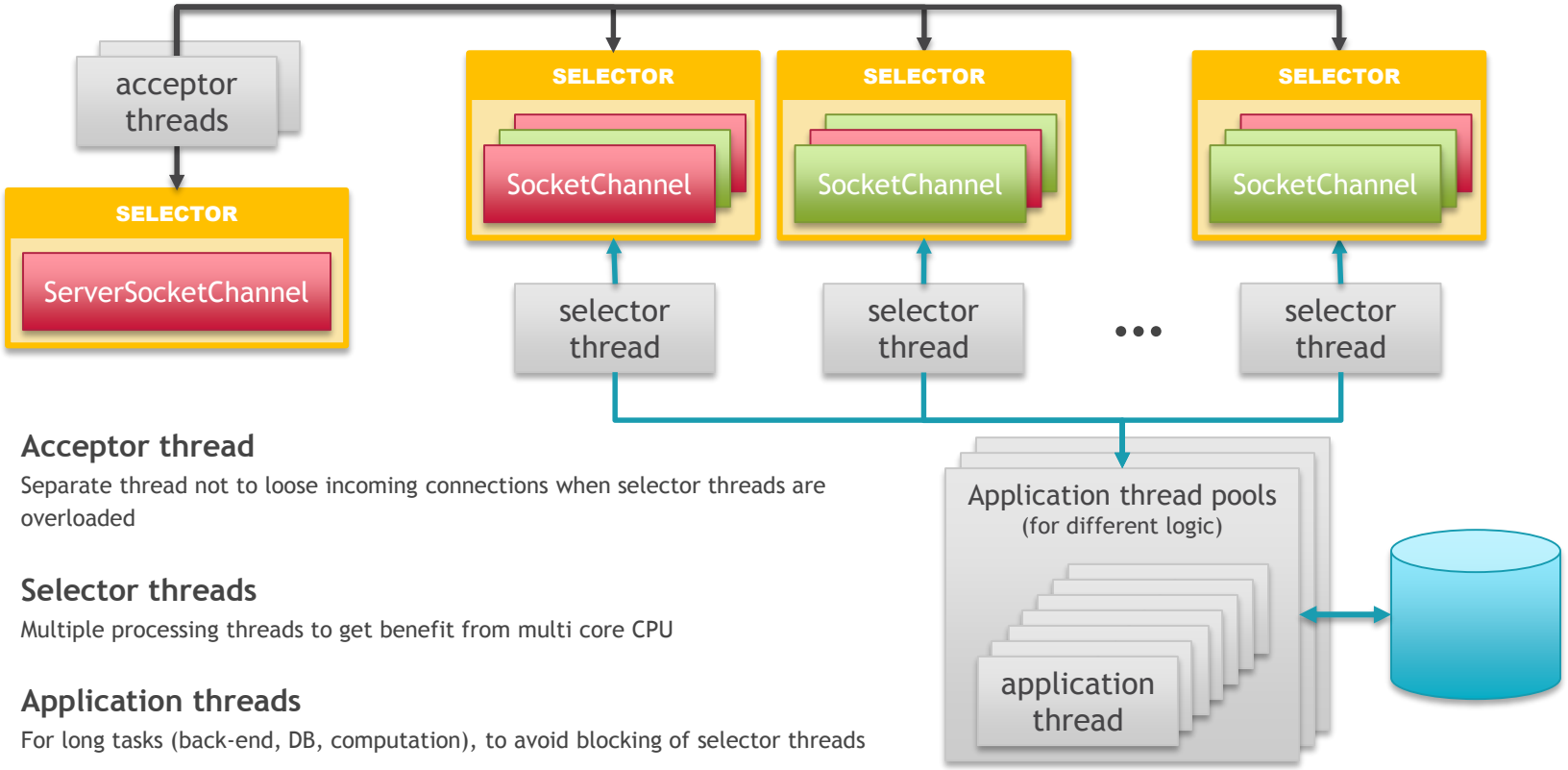
Basic NIO server architecture



Basic NIO server architecture



Production-ready NIO server



Acceptor thread

Separate thread not to loose incoming connections when selector threads are overloaded

Selector threads

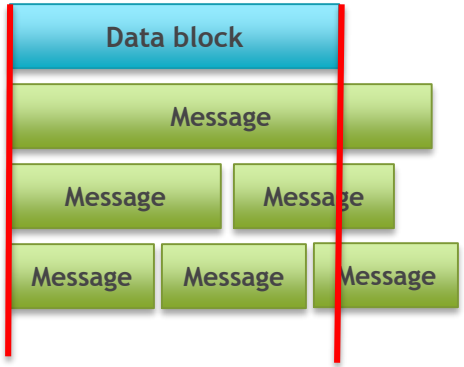
Multiple processing threads to get benefit from multi core CPU

Application threads

For long tasks (back-end, DB, computation), to avoid blocking of selector threads

Partial message receiving problem

Message reader splits/joins data blocks into messages



NIO conclusion

Advantages

Single thread for many connections

- less memory
- less context switching overhead

Drawbacks

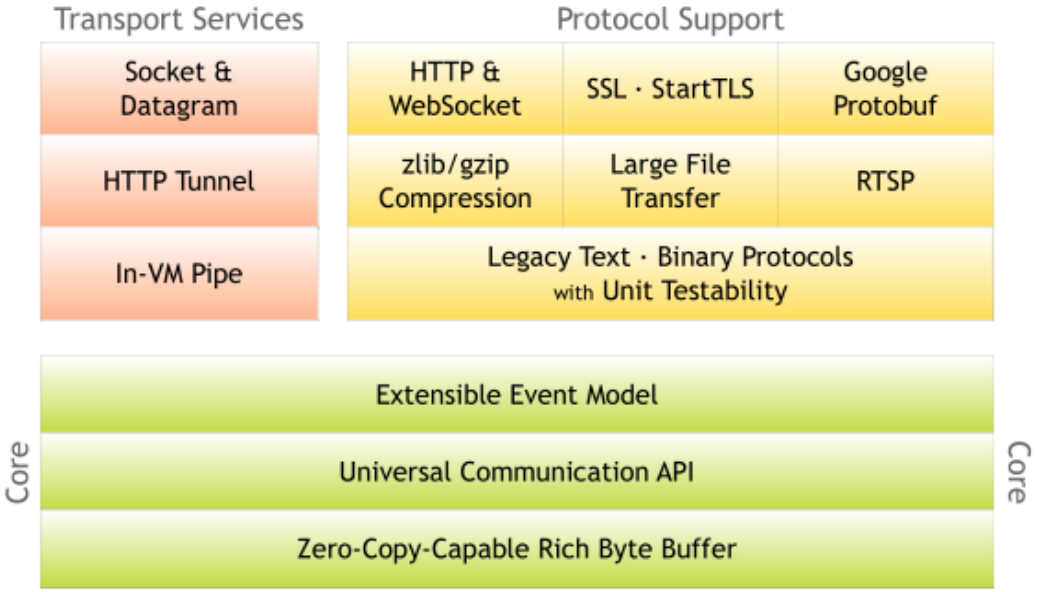
Complexity

- complex code
- handling partially received messages
- pain when we combine asynchronous and synchronous code – need different thread groups

THE WAY FROM PAIN TO PLEASURE

NETTY: ONE FRAMEWORK TO RULE THEM ALL

Netty: network application framework



Some projects using Netty

akka

Apache Spark

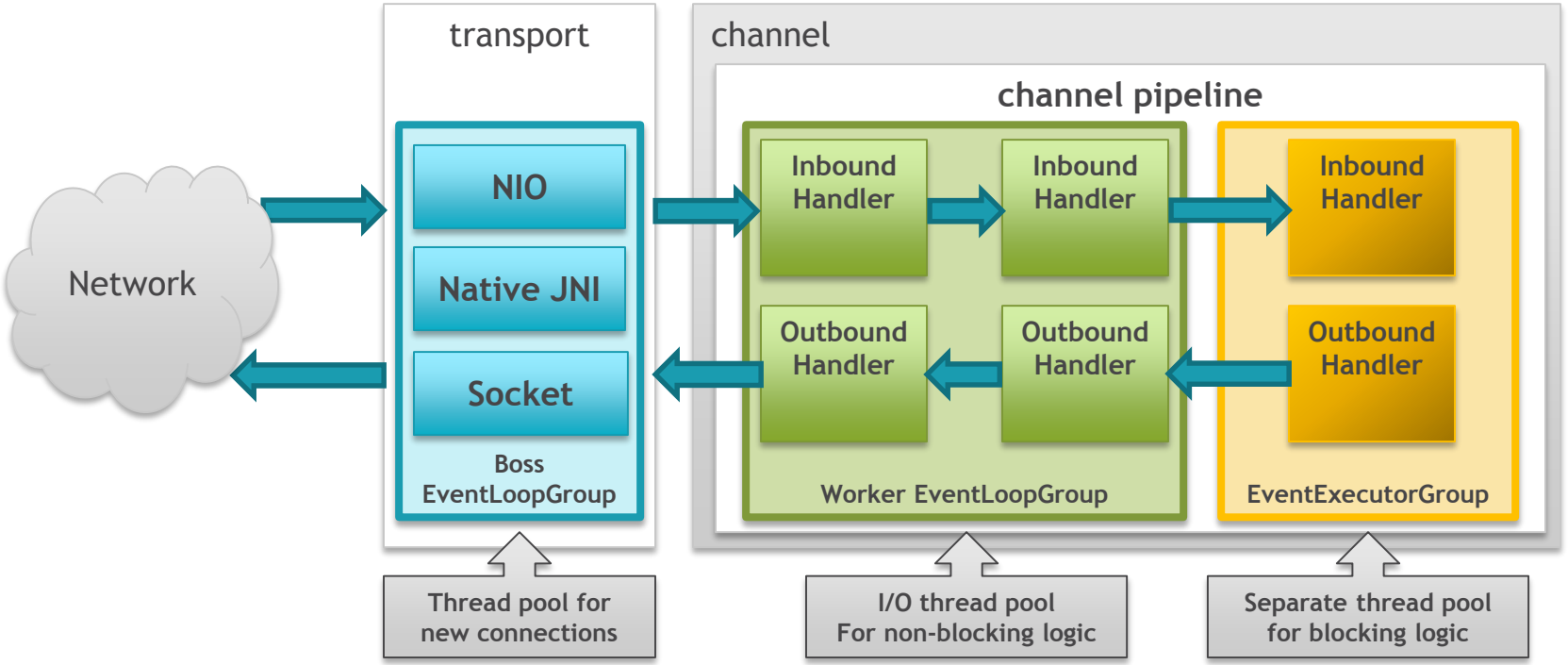
Apache Cassandra

Couchbase

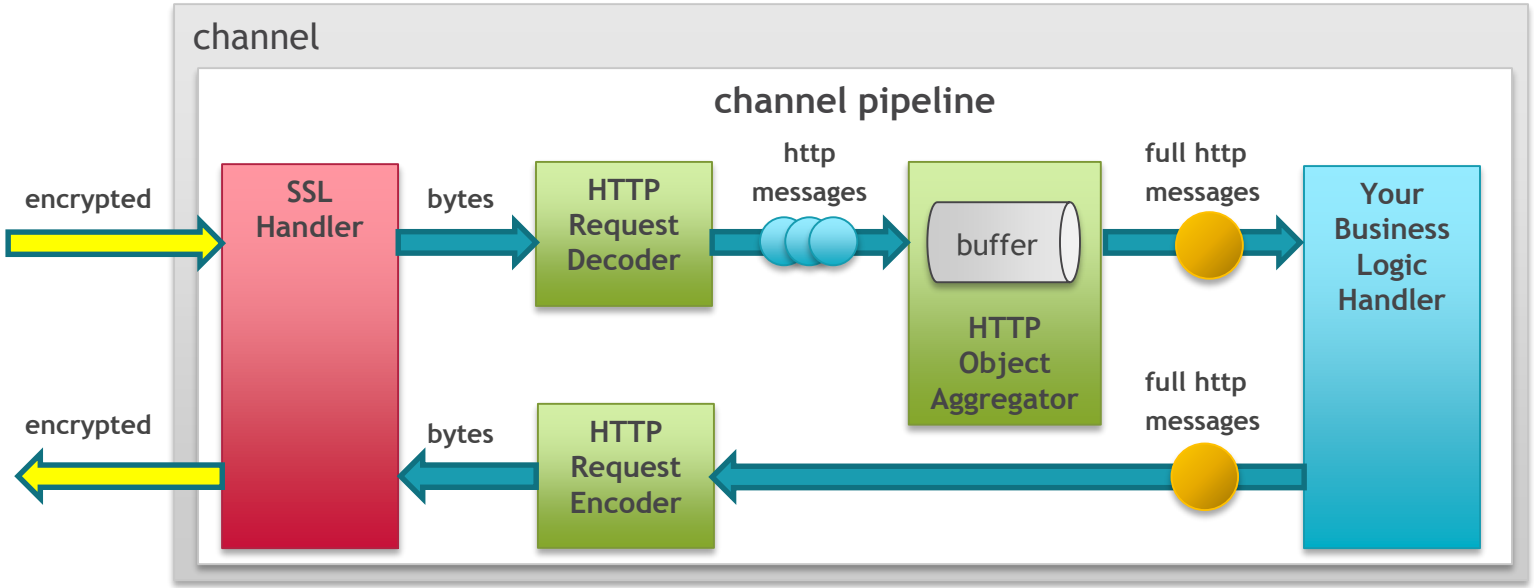
elasticsearch.

play

Netty server architecture

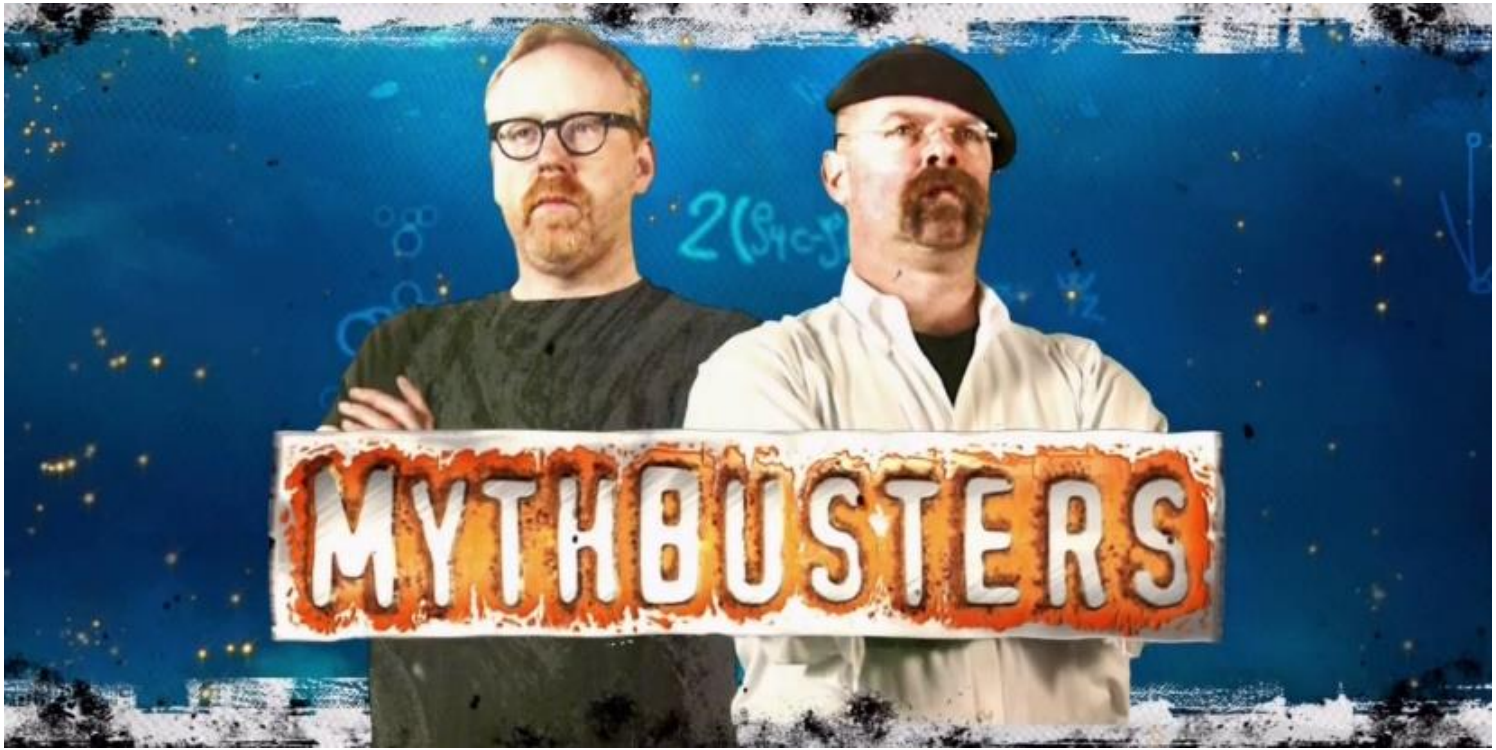


Netty handlers example




LIVE CODING EXAMPLE
NETTY SERVER

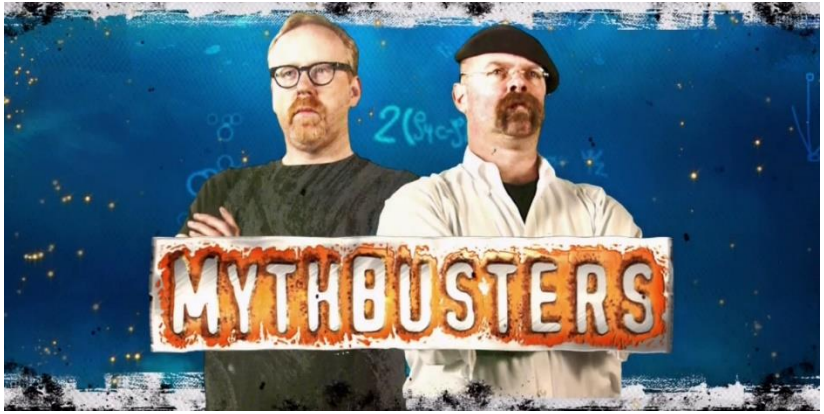
What is faster: IO or NIO?



What is faster: IO or NIO?

May be, nay be not *

1. … NIO is faster
2. … thread context switching is slow
3. … threads take up too much memory
4. … synchronization among huge number of threads will kill you
5. … thread per connection does not scale



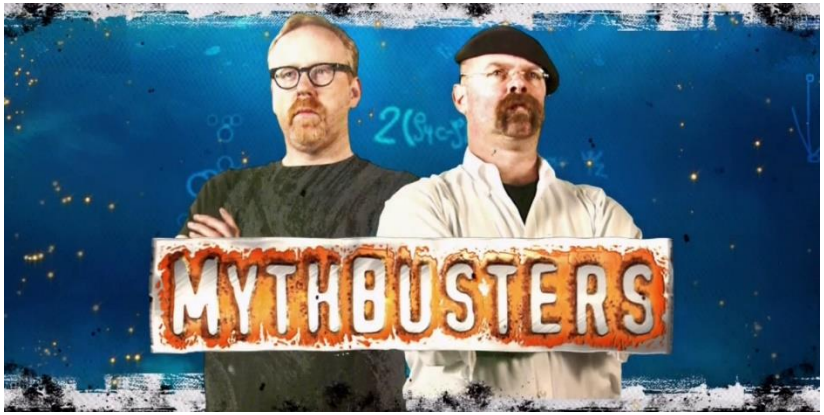
** Always point to holy-wars, see*

www.mailinator.com/tymaPaulMultithreaded.pdf

What is faster: IO or NIO?

May be, nay be not *

1. … NIO is faster
   Benchmark shows IO 25-30% faster*
2. … thread context switching is slow
   New threading library in Linux 2.6 (en.wikipedia.org/wiki/Native_POSIX_Thread_Library)
   Idle thread cost is near zero
   Context switching is much much faster
3. … threads take up too much memory
   Yes, it is, Linux (x64) default thread stack size (xss) = 1024kb
4. … synchronization among huge number of threads will kill you
   Non-blocking data structures which scale well
5. … thread per connection does not scale
   See 2-4



** Always point to holy-wars, see*
www.mailinator.com/tymaPaulMultithreaded.pdf

Conclusion

Use sockets*

- Small number of connections
- Huge data rate per connection
- **Synchronous world (procedure-based)**
  - Classic enterprise applications
  - Distributed XA transactions
  - Request-response protocols
  - Blocking database
  - Blocking RPC or web-service calls

- Bottleneck is not server, but blocking resource
- Make servers stateless and scale horizontally

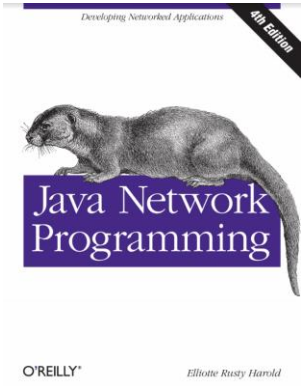
Use NIO*

- Huge number of connections
- Medium or slow data rate per connection
- **Asynchronous world (event-based)**
  - Non blocking database
  - Queues
  - Web-sockets
  - Callbacks, listeners
  - Data streaming

- Don’t use standard NIO, use frameworks like Netty
- Highly skilled team because of complex code

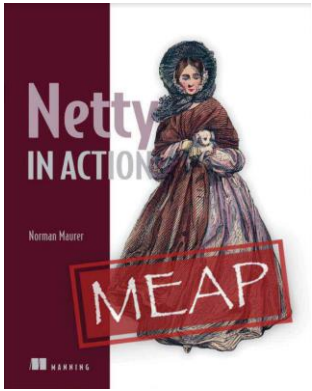
**Not a strict rule, always point to holy-wars*

BUY A BOOK, MAKE THEM RICH



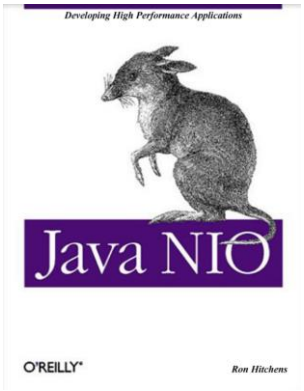
Java Network Programming, 4'th edition

O'REILLY
Elliotte Rusty Harold



Netty in Action

Manning
Norman Maurer



Java NIO

O'REILLY
Ron Hitchens

- **Code examples:** github.com/kslisenko/java-networking
- **NIO performance:** www.mailinator.com/tymaPaulMultithreaded.pdf
- **NIO tutorial:** tutorials.jenkov.com/java-nio/index.html
- **ITT 2015 - Heinz Kabutz - The Multi-threading, Non Blocking IO:** youtube.com/watch?v=uKc0Gx_lPsg
- **Netty - One Framework to rule them all by Norman Maurer:** youtube.com/watch?v=DKJ0w30M0vg
- **Scalable IO in Java:** http://gee.cs.oswego.edu/dl/cpjslides/nio.pdf

THANK YOU AND HAPPY HOLIDAYS!

QUESTIONS?

KANSTANTSIN_SLISENKA@EPAM.COM