

NVP – Nebenläufige und Verteilte Programme

Aufgabenblatt 11

Aufgabe 1

NIO-Frameworks Java-NIO wird von keinem Anwendungsprogrammierer und allen Java-basierten Server-Lösungen genutzt. Zwischen dem Anwendungsentwickler und I/O-Anweisungen liegen in der Regel mehrere Schichten an Software-Abstraktionen. Auf unterster Ebene und am weitesten von Anwendungsentwicklern entfernt finden sich Frameworks für effiziente und asynchrone Kommunikation, kurz “NIO-Frameworks”. Diese Frameworks bieten inzwischen mehr als nur eine passable Schnittstelle zu Kommunikations-Diensten auf Basis von NIO. Sie können in direkt verwendet werden um “leichtgewichtige” Anwendungen mit hohen Anforderungen an die I/O-Leistung zu entwickeln. In der Regel sind sie aber ein versteckter Bestandteil anwendungsnäherer “höherer” Frameworks, die von Anwendungen / Anwendungsentwicklern genutzt werden.

Einige bekannte Beispiele für solche “NIO-” oder “Basis-Frameworks” sind:

- Netty¹
- Grizzly²
- Apache Mina³
- Akka I/O⁴

Grizzly ist Bestandteil der Java-EE Referenz-Implementierung. Netty und Apache Mina haben einen sehr ähnlichen Funktionsumfang. Neben den genannten Frameworks gibt es unzählige andere, die mit der gleichen oder einer ähnlichen Zielsetzung entwickelt wurden.

Netty scheint dabei die größte Verbreitung zu haben. Es stellt das Herz vieler anwendungsnäherer Frameworks dar.

Aufgabe

1. Verschaffen Sie sich einen groben Überblick über die Funktionalität und das Einsatzgebiet von Netty. Werfen Sie dazu einen Blick auf die Internetpräsenz von Netty und sehen Sie das Video

<https://www.youtube.com/watch?v=DKJ0w30M0vg>

von Norman Maurer (ausschnittweise) an.

Wer ist Norman Maurer?

Wo wird Netty genutzt: Geben Sie einige “höhere Frameworks” oder Anwendungen an, die Netty nutzen.

2. *Jetty* kann leicht mit Netty verwechselt werden, ist aber etwas anderes. Was ist *Jetty*.
3. Netty kann mit folgende SBT-Abhängigkeitsdefinition in ein Projekt eingebunden werden:

```
libraryDependencies += "io.netty" % "netty-all" % "4.1.9.Final"
```

¹<https://netty.io>. Nicht zu verwechseln mit *Jetty*, das eine Stufe höher angesiedelt ist.

²<https://grizzly.java.net>

³<https://mina.apache.org>

⁴<http://doc.akka.io/docs/akka/current/scala/io.html> darin ist *Spray* (<http://spray.io>) aufgegangen.

Eine Anwendung, die eine Kopie einer Datei mit Hilfe von Netty erstellt ist:

```
import java.nio.channels.FileChannel
import java.nio.file.Paths
import java.nio.file.StandardOpenOption

import io.netty.buffer.PooledByteBufAllocator

object FileCopy_app {

  def main(args: Array[String]): Unit = {
    val src: FileChannel = FileChannel.open(
      Paths.get("/some/path/toa/file_1.txt"),
      StandardOpenOption.READ)
    val dest: FileChannel = FileChannel.open(
      Paths.get("/some/path/toanother/file_2.txt"),
      StandardOpenOption.WRITE, StandardOpenOption.CREATE)

    val alloc = PooledByteBufAllocator.DEFAULT
    val buf = alloc.directBuffer(256)

    var writerIndex = 0
    var readerIndex = 0
    var continue = true

    while (continue) {
      val bytesFromSrc = buf.writeBytes(src, writerIndex, buf.capacity())
      writerIndex = writerIndex + bytesFromSrc
      continue = bytesFromSrc > 0
      if (continue) {
        val bytesToDest = buf.readBytes(dest, readerIndex, bytesFromSrc)
        readerIndex = readerIndex + bytesToDest
      }
    }
    buf.release
  }
}
```

Vergleichen Sie diese Lösung mit einer reinen NIO-Lösung. Worin besteht der Unterschied, was bietet Netty in diesem Anwendungsfall?