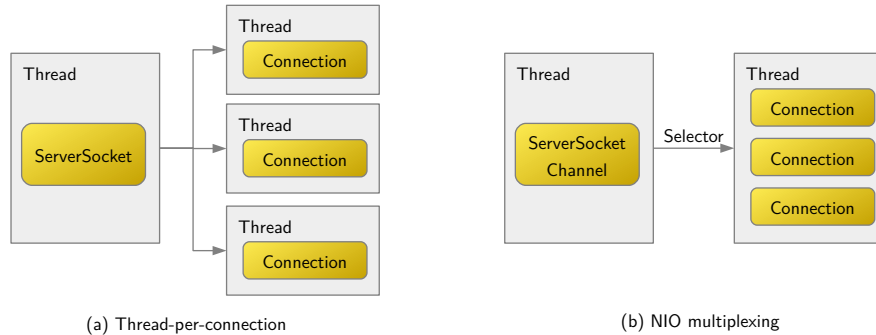


Performance Evaluation of Java IO Models



FAKULTÄT
FÜR INFORMATIK
Faculty of Informatics

184.714 Bachelor Thesis for Informatics and Business Informatics (10.0EC)



Motivation

Java NIO is an alternative IO API that was introduced in Java version 1.4 in 2002. Up until then, the standard way of dealing with IO was to work with streams and blocking IO operations. A thread that performs read or write operations on these streams has to wait until the operation is complete (blocking IO). For multi-connection systems, this means that each connection requires a separate thread. At the time, this imposed significant scalability limitations when faced with thousands of concurrent connections. In contrast, Java NIO uses the concept of IO multiplexing, where a single thread can operate on multiple connections using selectors and non-blocking IO operations. This significantly reduces the number of threads required to handle many connections.

However, this API was designed 15 years ago, when threads were considered an extremely expensive resource. Since then, multi-core architectures, operating systems and thread management have evolved significantly. There are legitimate doubts whether the non-blocking NIO model still outperforms thread-per-connection systems, even when scaling to tens of thousands of connections. Non-systematic benchmarks exist that give tentative evidence to this. Still, a lot of framework and libraries use NIO and advertise its benefits.

The aim of this work is to systematically evaluate the performance of Java's IO models (blocking, non-blocking, asynchronous) for different application requirements. Requirements are derived from state-of-the-art applications that use message-oriented middleware (e.g., MQTT message brokers and clients). The expected results are: a design of different evaluation experiments, a benchmarking framework to perform the experiments, and comprehensive results of different performance measures of the models.

Work Description

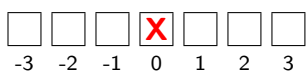
- Research of existing performance benchmarks of IO models
- Requirements analysis of distributed applications that use message-oriented middleware
- Development of experiments and benchmarking framework
- Presentation of performance evaluation results

What you will learn: performing empirical evaluations using micro-benchmarking tools (e.g., JMH); a deep understanding of IO models

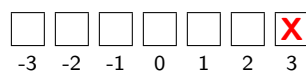
Further Information

Basic Requirements: Strong Java programming skills; good understanding of concurrency and socket programming

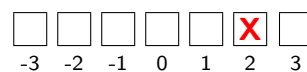
Contact: Thomas Rausch t.rausch@infosys.tuwien.ac.at



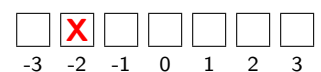
Conceptual (Analytical)



Empirical (Simulation)



Practical (Implementation)



Literature Work