

Master's Thesis

**A Comparison of Thread-based and Reactive Server Programming  
Models and Technologies**

Student: Marcel Homolka  
SKZ/Matr.Nr.: 01603779  
Email: marcel.homolka@gmx.at  
Advisor: a.Univ.-Prof. Dr. Herbert Prähofer  
Start date: October 2020

**a.Univ.Prof. Dr. Herbert Prähofer**  
Institute for System Software

T +43 732 2468 4352  
F +43 732 2468 4345  
herbert.praehofer@jku.at

Goal of the master thesis is a comparison of different programming models and technologies for implementing server systems. In particular, the thesis should work out the differences between classical thread-based technologies and new reactive technologies.

**Technologies to compare**

Technologies to compare are on the one side classical thread-based technologies, e.g., Servlets, and on the other side the newly appearing reactive technologies.

Thread-based:

- Java Servlets and JSP

Reactive technologies:

- Node.js
- Vert.x
- RxJava
- Akka

**Case studies**

The comparison should be done based on two example server systems, which are rather different in the way how the clients interact with the server. One is session-based and the other message-based.

*Session-based interactions:*

Session-based servers are characterized by a longer interaction between the clients and the server. The client undergo a longer process where a state for the client has to be maintained by the server

Typical example system is a Web shop

*Message-based interactions:*

The client does not undergo a longer-lasting process but clients sporadically send short messages to the server. No state for the client is maintained at the server between client requests. Here, the main challenge is the number of clients and the number of messages sent per time unit.

Typical example system is a message-server like What's app.

**Evaluation:**

The technologies should be compared regarding number of clients, which can simultaneously be connected to the server and the throughput in the form of number of interactions/messages from clients.

Further, the candidate should do a qualitative comparison of the programming models. No formal evaluation is needed, but a more subjective judgement by the author regarding effort for implementation, readability and maintainability of programs should be done.