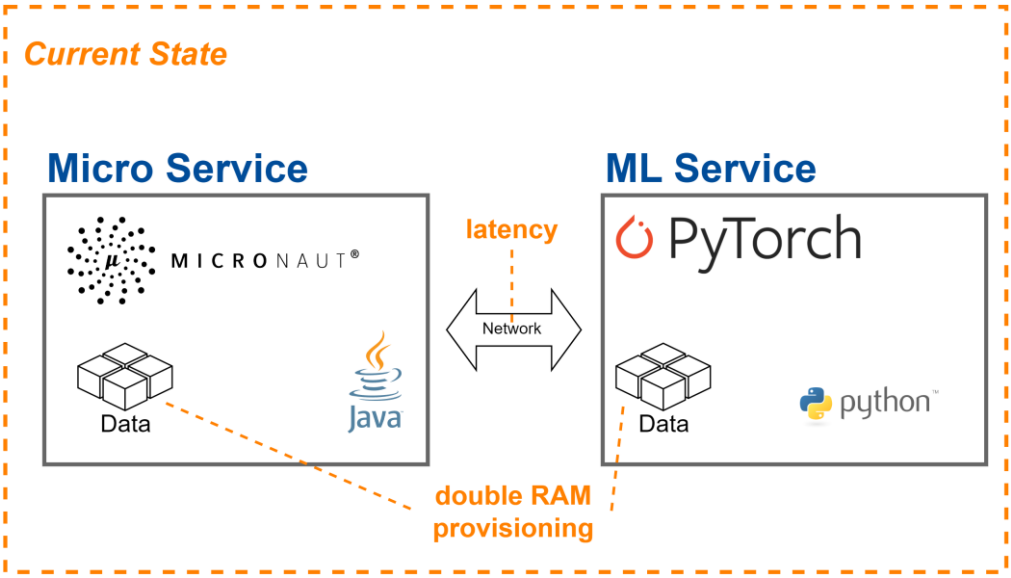
Building Machine Learning Macro-Services with GraalPy and Micronaut

Master Project Proposal, Winter Term 2023/24

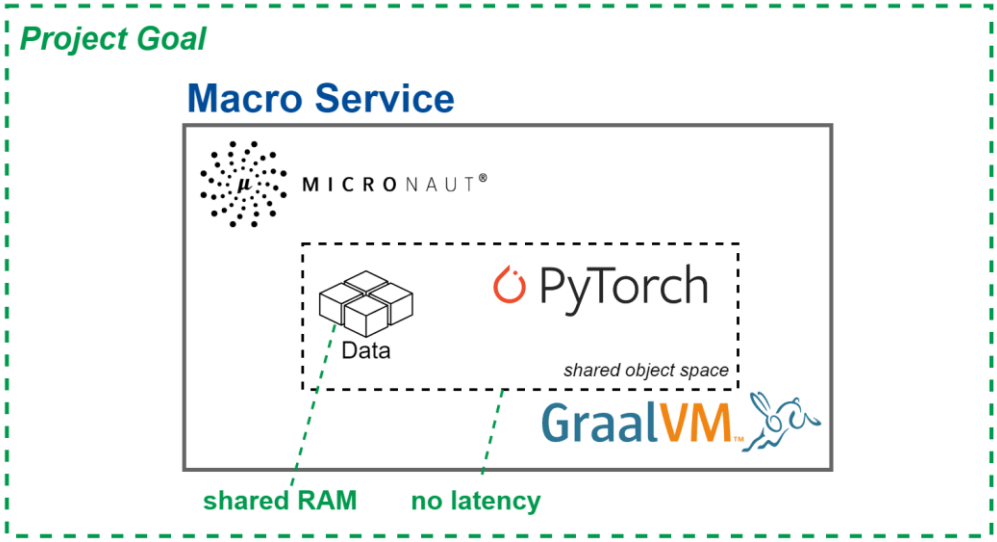
Software Architecture Group, Prof. Dr. Robert Hirschfeld

In this project, students will combine the power of Python ML libraries with the Java-based Micronaut microservice framework to build novel machine learning macro-services that are easy to deploy to different clouds.



Background

Python is known for its rich ecosystem of data science (DS) and machine learning (ML) libraries. The GraalPy runtime is a Python implementation with deep integration into Java and brings first-class Python DS and ML libraries to Java projects. Powered by the open source Graal just-in-time (JIT) compiler and GraalVM Native Image ahead-of-time (AOT) compilation, GraalPy is fast and compatible with many modern Python and Java libraries. Micronaut is a Java microservice framework that works well with GraalVM. The GraalVM team provides Micronaut tooling for Visual Studio Code that makes it easy to build and deploy Micronaut applications to multiple cloud providers via Graal Cloud Native. The tools also make it easy to use GraalVM Native Image to create single binaries from Micronaut applications. These are self-contained, do not need Java to run, and have fast startup and low footprint.



Details

State-of-the-art applications mixing languages like this are usually built with microservices or multiple processes. This incurs a large overhead when dealing with the amount of data in DS and ML contexts, and the services become bottlenecked by the I/O pushing the data between Java and Python. GraalVM can fix this by executing both Java and Python in the same VM. Since this is very new technology, the user experience and tooling of mixed language projects is still in its infancy. In this master's project, students will investigate how Java programmers wanting to use Python libraries for machine learning may integrate these into their projects. How is the development lifecycle? How are interfaces between Java and Python structured? Which tools are used to write and execute the unit tests? How does GraalVM Native Image work in this configuration? How can this be deployed in the cloud? Micronaut with its first-class GraalVM tooling support and the PyTorch ML library will serve as a basis for the project. Students can influence the direction of the project and the concrete use cases to look at. Members of the GraalVM team at Oracle Labs collaborate with the SWA research group to act as customers and provide technical assistance.

Further Reading

[1] High-performance Modern Python (*https://graalvm.org/python*), 2023

[2] GraalVM Native Image (*https://www.graalvm.org/native-image/*), 2023

[3] Use Python and R in your Java applications with GraalVM (*https://blogs.oracle.com/javamagazine/post/java-graalvm-polyglot-python-r*), 2022

[4] A modern, JVM-based, full-stack framework for building modular, easily testable microservice and serverless applications (*https://micronaut.io), 2023*

[5] Graal Cloud Native (*https://www.graal.cloud/gcn/*), 2023

Contact

Prof. Dr. Robert Hirschfeld, Dr. Jens Lincke,
Dr. Fabio Niephaus, Dr. Tim Felgentreff