

発表概要

Java 静的コンパイラを用いた Quarkus フレームワークの性能評価

伊澤 侑祐^{2,a)} 堀江 倫大^{1,b)} 緒方 一則^{1,c)} 千葉 立寛^{1,d)}

2021年11月1日発表

クラウド環境において、アプリケーションをコンテナ化し配置・実行・管理するモデルが主流である。Red Hat により活発に OSS 開発が進められている Quarkus は、マイクロサービスやサーバレス・アプリケーション開発のための基盤フレームワークである。Quarkus は、GraalVM を用いて Java プログラムを事前にネイティブ・バイナリへ変換することにより、Java 仮想機械使用時のメモリ消費量や実行時間のオーバーヘッドを回避できる。さらに、依存性注入の解決を事前コンパイル時に行い、実行時の負荷を削減することも可能である。一方、Java 仮想機械を利用した従来の Java アプリケーション実行も可能であり、実行時情報を利用した高度な実行時最適化を適用できる。本研究は、ワークロードごとにその特性に基づいてネイティブ・イメージ実行と Java 仮想機械実行を使い分ける新たな実行モデルの実現を目指す。本発表では、その第 1 歩として、DaCapo ベンチマークを用いてネイティブ・イメージと Java 仮想機械のどちらが適しているかワークロード特性に基づき調査する。

Presentation Abstract

An Evaluation of the Quarkus Framework with a Java Static Compiler

YUSUKE IZAWA^{2,a)} MICHISHIRO HORIE^{1,b)} KAZUNORI OGATA^{1,c)} TATSUHIRO CHIBA^{1,d)}

Presented: November 1, 2021

In cloud-based applications, developers deploy, execute and manage their implementations as a container. Quarkus, an actively developed OSS by Red Hat, is a Java framework to create effective and efficient container-based Java applications. Quarkus has two running options; a native binary and Java virtual machine (JVM). Executing as a native binary reduces memory consumption as well as avoiding a just-in-time (JIT) compiler's warm-up time. It also removes runtime overhead by resolving dependency injections at the compilation time. On the other hand, executing on a JVM utilizes runtime information to produce faster native code with a JIT compiler. Our motivation is to realize a new execution model which exploits the two advantages of native binary and JVM executions. As a first step, in this presentation, we investigate and organize their performance characteristics from the viewpoint of workload properties.

This is the abstract of an unrefereed presentation, and it should not preclude subsequent publication.

- ¹ 日本 IBM 株式会社東京基礎研究所
IBM Research - Tokyo, Chuo, Tokyo 103-8510, Japan
- ² 東京工業大学情報理工学院数理・計算科学系
Department of Mathematical and Computing Science, Tokyo
Institute of Technology, Meguro, Tokyo 152-8550, Japan
- a) izawa@prg.is.titech.ac.jp
- b) horie@jp.ibm.com
- c) ogatak@jp.ibm.com
- d) chiba@jp.ibm.com