

บทคัดย่อ

งานวิจัยนี้ศึกษาการพัฒนาคอมไพเลอร์สำหรับแปลงโปรแกรมภาษาจาวาให้เป็นโปรแกรมที่ใช้ชุดคำสั่ง invokedynamic เพื่อเพิ่มคุณสมบัติความพลวัตให้กับโปรแกรม โดยทำการดัดแปลงคอมไพเลอร์ของจาวาใน OpenJDK 1.7 ให้สร้างเมธอดเริ่มต้นที่ทำการโยนไปหาตัวจัดการเมธอดและวัตถุคอลไซต์ซึ่งเชื่อมต่อกับเมธอดจริง แทนการเชื่อมต่อกับเมธอดจริงโดยตรง โดยเสนอกฎการแปลงในรูปแบบตามแคลคูลัสของ Featureweight Java

จากผลงานวิจัย พบว่าสามารถพัฒนาคอมไพเลอร์ที่แปลงโปรแกรมให้ใช้ชุดคำสั่ง invokedynamic ได้สำเร็จโดยทดสอบกับชุดตัววัด SciMark 2.0 และได้ผลการทำงานถูกต้อง 100% ด้วยชุดตัววัดดังกล่าว สำหรับประเด็นด้านประสิทธิภาพพบว่าโปรแกรมที่ใช้ชุดคำสั่ง invokedynamic ส่วนใหญ่มีประสิทธิภาพอยู่ในระดับเกิน 20% ของโปรแกรมดั้งเดิม โดยพบว่าโปรแกรมที่มีการคำนวณซับซ้อนจะทำให้ JVM ไม่สามารถดึงประสิทธิภาพของโปรแกรมที่ใช้ชุดคำสั่ง invokedynamic ได้เต็มที่ ซึ่งเป็นประเด็นที่สามารถปรับปรุง JVM ได้ในรุ่นต่อ ๆ ไป

Abstract

The research work described in this report studied a development of a compiler for translating a Java binary program into a program that utilizes invokedynamic instructions. The benefit of adding invokedynamic instructions is for increasing the dynamic property for the compiled program. The work tweaked a compiler in OpenJDK 1.7 to make it generate Bootstrap Method that allow linking Method Handles with Callsite Objects which connected with real methods. This work explain transformation rules using Featherweight Java calculus.

From the experimental results, the compiler can successfully translate programs from the SciMark 2.0 suite to be invokedynamic implanted and they work 100% correctly. From the performance perspective, the translated programs have performance around 20% compared to the original benchmark programs. There is an observation that the JVM could not optimize a complex invokedynamic program well enough, which is a for improving JVM in the future.



มหาวิทยาลัยเทคโนโลยีสุรนารี