
発表概要

Java におけるコード配置の動的最適化

田中 慎司[†] 山田 哲靖^{††} 八尾 宏[†]

現在、Web アプリケーションなどの分野における Java 言語の採用が進んでいる。Java 言語の利点は、主にアプリケーションの移植性と生産性にあるが、Java 言語の処理系である JavaVM の問題として、性能があげられることが多く、重要な課題となっている。JavaVM は、プログラムの実行中にバイトコードからネイティブコードに動的にコンパイルが実行されるなど、動的な最適化の適用が容易となっている。動的な最適化では、ソースコードなどの静的な情報だけではなく、実行時に取得するプロファイルによって、さらなる最適化が可能である。本発表では、JavaVM の高速化のために、動的コンパイル時に実行頻度の低いコードを他のコードと分離することで、実際のアプリケーション性能に大きく影響するキャッシュ利用効率を向上させる手法を提案する。本手法では、実行頻度の低いコードを発見するために、各メソッドの Basic Block ごとの実行回数、およびメソッド呼出しの呼出し元と呼出し先を記録する。それぞれを、一定時間取得し、Basic Block の実行回数、およびメソッドの呼出し回数によって、それぞれの最適な配置と分割方針を判断する。その後、メソッドの再コンパイルにより、再配置を実行することで最適配置を実現する。本手法のプロトタイプ実装による評価結果より、J2EE のベンチマークで 14.3%、SPEC jbb2000 で 2.2% の性能改善効果を得た。

Dynamic Optimization for Code Positioning for Java

SHINJI TANAKA,[†] TETSUYASU YAMADA^{††} and HIROSHI YAO[†]

Recently, Java language is spreading over web application servers and other servers. Advantage of Java language are mainly portability and productivity of applications. But performance of JavaVM, which is execution environment of Java language, is often focused as problem. It is easy to apply dynamic optimization to JavaVM, such as dynamic compilation of binary codes to native codes. Dynamic optimization can use static information like source codes and dynamic information like online profiles. In this presentation, we propose a method to divide frequently executed codes from infrequently executed codes. The method can improve cache hit rates, which directly affects system performance. With this method, JavaVM counts the number of executions of basic blocks of all methods and the caller and the callee of all method calls. After the profiling period, JavaVM determines positions of basic blocks and methods based on the profiles. And finally JavaVM recompile every methods for re-positioning them. From the result of the evaluation of the prototype implementation, this method can improve the performance of an benchmark of J2EE by 14.3%, and the performance of SPEC jbb2000 by 2.2%.

(平成 17 年 10 月 12 日発表)

[†] 日本電信電話株式会社, ネットワークサービスシステム研究所
Network Service System Laboratory, NTT

^{††} NTT ソフトウェア株式会社
NTT Software