
発表概要

Java 言語上の細粒度マルチスレッドフレームワーク における問題点の考察

西川 健一[†] 脇田 建[†] 佐々政孝[†]

並列性をよりシンプルかつポータブルに、そしてより効率的に表現することを可能にする並列プログラミングのためのモデルの重要性が増している。一般に、細粒度スレッドパッケージは、膨大な数のスレッド生成とプロセス数を意識せずにプログラムを記述することを可能にする。Doug Lea の FJTask は Cilk の work-stealing アルゴリズムを Java で実装した Java 言語上の細粒度マルチスレッドフレームワークである。従来の Cilk 等の C を基盤としたものと比較して、プラットフォームからの独立性の高いコードが期待できる。しかし、我々の実験によると FJTask には、いくつかの問題点があることが分かった。細粒度のスレッド生成をオブジェクトの生成に置き換えているため、ごみ集め処理のオーバーヘッドが増大する等の性能上の問題を引き起こしている。本研究では、FJTask における性能上の問題点を考察し、その改善案を提案する。オブジェクトを再利用する仕組みを取り入れ、ごみ集め処理のオーバーヘッドを削減し、実行時間および台数効果を改善した。特に Classic VM を搭載する JVM において、その効果は顕著であった。

Problems of the Fine-grain Multithread Framework on Java

KENICHI NISHIKAWA,[†] KEN WAKITA[†] and MASATAKA SASSA[†]

Generally, fine-grain thread packages are capable to produce a huge number of threads and mapping them to massively parallel processors. This feature releases the programmer from management of thousands of processing units offered by the massively parallel machines. Doug Lea's FJTask package is a fine-grain multi-thread framework that implements the work-stealing algorithm firstly implemented for the Cilk system. An advantage of FJTask package in comparison with the previous work is the architectural independence from the underlying platforms. However, the effectiveness in terms of actual execution efficiency is questionable. The authors have measured the execution efficiency of the FJTask package through a number of benchmark programs and identified some of the problems with the design and implementation of FJTask. Most importantly, because each fine-grain thread is represented by a heap-allocated object, massively parallel execution can incur enormous overhead to the memory allocator and the garbage collector of the Java runtime system. The bottleneck of FJTask is examined in detail and an alternative implementation is proposed. In the proposed implementation dead objects are found and reused so that most of the memory management overhead are eliminated. Our proposal is compared with the Lea's original proposal on a variety of platforms.

(平成 13 年 7 月 27 日発表)

[†] 東京工業大学大学院数理・計算科学専攻
Department of Mathematical and Computer Sciences,
Tokyo Institute of Technology