

発表概要

ノンブロッキング型動的負荷分散を用いた
並列ガーベージコレクション方式前田 宗 則[†] 中村 実[†] 小沢 年 弘[†]

本発表では、ノンブロッキング型ワークスチーリング法に基づく動的負荷分散機構の実装について報告する。本機構は我々の実装した並列ガーベージコレクション (GC) の中核となるものである。ワークスチーリング法に基づく動的負荷分散方式とは、並列処理をワークと呼ばれる部分処理に分割し、実行スレッドは各々のワークキューに処理待ちのワークを保持する。ワーク割付けの均等化は、実行すべきワークが不足したスレッドが自律的に他のスレッドのワークキューからワークを奪うものである。本方式では、スチール操作を含むキューの排他操作が性能面で支配的になる。さらに、キュー操作を OS の提供するロックを用いて実装するならば、ロックの競合や遅延などにより大幅な性能劣化が生じる場合がある。Arora らの Deque アルゴリズムは、スチール操作をアトミック命令を用いてノンブロッキング化したものであり、実行スレッドは自らのワークキューからのワークの取出しはつねに定数時間で完了することができるという長所を持つ。他方、1 回のスチール操作でただか 1 つのワークしか横取りできないという短所を持つ。Hendler らは、これを 1 回のスチール操作で最大半分のワークを相手先から横取り可能とする拡張を提案している。ただし、スチール操作は横取りするワーク量に比例した実行時間を必要とする。我々は、Arora らの方式を基礎とし、1 回のスチール操作で一定数のワークを定数時間で横取りする方式を提案する。本発表では、これら 3 つの方式を比較し、並列 GC における性能を示す。

Parallel Garbage Collection with Non-Blocking Work-Stealing Queue

MUNENORI MAEDA,[†] MINORU NAKAMURA[†] and TOSHIHIRO OZAWA[†]

Work stealing is the one of well-known schemes in load balancing technology: Multiple threads have their own work queues and each thread push and pop items from its work queue. When a queue becomes empty, the owner thread tries to steal items from other threads. For fine-grained multi-threaded programs, the cost of queue operations with mutual exclusion among threads is dominant in performance. In recent years, non-blocking work-stealing algorithm proposed by Arora et al. are becoming popular as the scheme for the shared memory multiprocessor. The algorithm provides efficient queue operations by using atomic instruction such as `compare_and_swap` (CAS) without using lock operation. It enables to make a thread push and pop an item from its own non-empty queue with no atomic instructions in most cases, and to get at most one item per steal operation in constant time by using CAS. Hendler et al. extend the algorithm to steal half amount of items per steal operations in proportional time to amount. We propose an algorithm to steal a packed items in constant time, based on Arora's. In this talk, we present our extension in detail. The work stealing mechanism is adopted to our parallel garbage collector (GC). We show the performance comparisons with these non-blocking algorithms under the configuration in our parallel GC.

(平成 16 年 7 月 30 日発表)

[†] 株式会社富士通研究所
FUJITSU Laboratories LTD.