

発表概要

3種類の並列キュー計算モデルの基本特性

クトゥルク ハルチャム[†] 川田 宗太郎[†]
 ベン アブデザゼック[†] 曾 和 将 容[†]

キュー計算モデルは FIFO の高速メモリを途中結果の格納用に用いる計算モデルである。従来のキュー計算モデルは、クロスアークや命令ホール問題や参照問題などがあり、プログラムの書き方に対する様々な制約がプログラミングを複雑にしていた。我々はこの問題を解くために、「消費順序遵守型キュー計算モデル」、「生産順序遵守型キュー計算モデル」を提案した。またそれともなって従来のキュー計算モデルを「生産消費順序遵守型キュー計算モデル」と名付けた。キュー計算モデルは並列処理に向いているので、これを生かすために、これらの計算モデルを並列処理可能なように拡張し、プロセッサの設計やコンパイラの構築などいろいろな研究を行ってきた。しかし、これまでこれらのモデルの基本的な性質や基本特性が明らかにされていなかった。本発表では、これらの問題の解決法とプログラミングについて述べ、ベンチマークプログラムによって3種類の並列モデルの基本的な特性（命令数、実行ステップ数、最大並列度、平均並列度、必要となる最大のキューのワード数など）を明らかにする。その結果生産型並列キュー計算モデルが、種々の問題が解決できる、同じデータを複数回利用できる、命令レベル並列性を多く含むプログラムに向いているなど、ほかの2種類のモデルよりも優れた特性を持っていることが明らかになった。

Fundamental Characteristics of Three Types of Parallel Queue Computation Models

HALCHAM KUTLUK,[†] SOTARO KAWATA,[†] ABDERAZEK BEN[†]
 and MASAHIRO SOWA[†]

Queue can be used to store intermediate calculation results into a First In First Out data structure. There are some problems to make programming on a FIFO, these problems are named “Cross arc problem”, “Instruction hole problem” and “compare problem”. In our previous research on the queue computational model we have found restrictions on writing queue programs and we have proposed some solutions: Consumer Order Preserving Queue computational model, Producer Order Preserving Queue computational model, and the original queue computational model combining both the above mentioned models is called Producer Consumer Order Preserving model. Furthermore, our research includes the development of the architecture and optimizing compiler for a parallel queue processor capable to execute effectively programs written for the queue computational model. Up until now, none of the three previously mentioned queue models has been described theoretically. The purpose of this presentation is to describe the characteristics (in terms of number of instructions, program size, execution cycles and degree of parallelism) of the three models through the analysis of benchmark programs and comparing the performance of the models. From our analysis, we found that the “Produced” computing model is better than the other models in terms of program size, instruction level parallelism and execution speed.

(平成 18 年 10 月 12 日発表)

[†] 電気通信大学大学院情報システム研究科
 Department of Information System, University of
 Electro-Communications