



Title	マスター・ワーカ型並列プログラムを高速に実行するためのコンパイラ支援に関する研究
Author(s)	水谷, 泰治
Citation	大阪大学, 2005, 博士論文
Version Type	VoR
URL	https://hdl.handle.net/11094/2316
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

氏名	みず 谷泰治
博士の専攻分野の名称	博士(情報科学)
学位記番号	第19611号
学位授与年月日	平成17年3月25日
学位授与の要件	学位規則第4条第1項該当 情報科学研究科コンピュータサイエンス専攻
学位論文名	マスター・ワーカ型並列プログラムを高速に実行するためのコンパイラ支援に関する研究
論文審査委員	(主査) 教授 萩原 兼一 (副査) 教授 増澤 利光 教授 松田 秀雄

論文内容の要旨

これまでに、並列再帰(再帰処理の並列化)を扱える並列化コンパイラが開発されている。これらのコンパイラは、再帰処理の並列化方式として、負荷分散を用いない方式、あるいは動的負荷分散が可能なマスター・ワーカ(MW)方式の一方を採用している。並列再帰プログラムを高速に実行するためには、MW方式の負荷分散効果とオーバヘッドのトレードオフを考慮して並列化方式を選び、その並列化方式を扱うコンパイラを用いて開発する必要がある。また、MW型並列プログラムを高速に実行するためには、マスターの過負荷による性能低下を生じさせない実行パラメータ(ワーカ数、マスター数、タスクの一括割当数)の値を検出し、その値を用いて実行する必要がある。

本論文では、この開発と検出の支援を目指し、(1)並列化方式を指定できるコンパイラによる、並列再帰プログラムの開発の支援、および(2)MW型並列プログラムの性能予測による、高速実行できる実行パラメータ値の検出の支援、に取り組んでいる。

まず(1)では、どの並列化方式(MW方式か負荷分散を用いない方式)が並列再帰プログラムの実行時間をより短くできるかを、多くの場合、開発者自身が予想できる点に着目している。また、提案しているコンパイラでは、過剰な並列化を抑制するための条件(並列化条件)およびMW方式におけるマスター数の指定も可能としている。そして、マージソート、クイックソート、およびn女王問題を用いて実験を行い、並列化方式と並列化条件の違いによって、実行時間にそれぞれ最大25%および最大77%の差が生じることを示している。また、並列化方式および並列化条件の指定方針について考察し、その方針に従うことで実行時間を短くできることを示している。さらに、マスター数の指定によって実行時間を短くできることも示している。

次に(2)では、マスターの過負荷において従来手法が高精度に性能予測できない原因を分析し、性能予測の高精度化のための考慮点として、(D1)並列計算モデルの利用による予測オーバヘッドの低減、(D2)並列計算モデルの拡張によるマスターの通信オーバヘッドのモデル化、および(D3)プログラム実行時に決まる挙動の再現の3点を示している。また、これらの考慮点に基づく性能予測の評価実験を行っており、従来手法ではマスターが過負荷時の予測誤差は最大42%であるのに対し、考慮点に基づく性能予測では最大10%の予測誤差であることを示している。

さらに、(2)における性能予測の高速化手法を提案している。提案手法では、一部のタスクの実行時間から残りのタスクの実行時間を線形補間によって推測することで、直接実行部分を削減し、性能予測の高速化を実現している。ま

た、この推測において、高い予測精度を維持するために、タスクの割当順に従って個々のタスクの実行時間も正確に推測する必要性を述べている。さらに、実験によって、提案手法による性能予測は従来手法より 1.7 倍以上高速であり、実測との誤差は 7 % 以下であることを示している。

論文審査の結果の要旨

近年、PC クラスタやグリッド計算環境の普及により、分散メモリ環境の多様化が進んでいる。このような環境において、並列プログラムを効率良く開発し、高速に実行できることは重要である。本論文では、この開発と実行の支援を目的とし、(1)並列化方式を指定できるコンパイラによる、並列再帰プログラムの開発の支援、および(2)マスタ・ワーカ (MW) 型並列プログラムの性能予測による、高速実行できる実行パラメータ値の検出の支援に取り組んでいる。

まず(1)では、再帰処理の実行時間を短くできる並列化方式を、多くの場合、開発者が予想できる点に着目している。そして、実験によって提案コンパイラの有用性を確認している。また、並列化方式および並列化条件の指定方針について考察し、その方針によって並列再帰プログラムの実行時間を短くできることを示している。さらに、MW 方式におけるマスタ数の指定によって実行時間を短くできることも示している。これらの成果は、アルゴリズムを構築する上で重要な技法である再帰を容易に並列化でき、開発者に対して並列プログラムの設計の幅を広げた点で有用といえる。

次に(2)では、MW 型並列プログラムの高精度な性能予測を実現するための考慮点として、(D1) 並列計算モデルの利用による予測オーバヘッドの低減、(D2) 並列計算モデルの拡張によるマスタの通信オーバヘッドのモデル化、および (D3) プログラム実行時に決まる挙動の再現の 3 点を示している。また、実験によって、従来手法での予測誤差は最大 42 % であるのに対し、考慮点に基づく性能予測によって最大 10 % という良い精度で予測できることを示している。

さらに、(2)における性能予測の高速化手法を提案している。提案手法では、一部のタスクの実行時間から残りのタスクの実行時間を線形補間によって推測することで、直接実行部分を削減し、性能予測の高速化を実現している。さらに、実験によって、提案手法の予測速度は従来手法より 1.7 倍以上であり、実測と予測の誤差は 7 % 以下であることを示している。この結果は、従来手法と同程度に良い予測精度を保ちつつ高速に性能予測できることを示しており、提案手法は有用であるといえる。

以上の研究成果は、多様化が進む近年の分散メモリ環境において、並列プログラムの開発者に対して並列プログラムの開発を容易にし、また、MW 型並列プログラムの高速な実行を支援できる点で有用であり、博士（情報科学）の学位論文として価値のあるものと認める。