

博士論文要旨

論文題名：データ並列タスクのマルチコア スケジューリングアルゴリズム

立命館大学大学院理工学研究科
電子システム専攻博士課程後期課程

リュウ ヨウ

LIU Yang

年々高まる計算性能への要求を満たすため、汎用プロセッサおよび組込みプロセッサの CPU コア数は増加の一途を辿っている。そのような高い並列性を持つプロセッサの能力を最大限に活用するためには、タスクスケジューリングと呼ばれるソフトウェア技術が非常に重要である。

ソフトウェアの並列処理は、タスク並列処理とデータ並列処理の 2 種類に大別される。タスク並列処理は、複数のコア上で異なるタスクを並列に実行する。一方、データ並列処理は、複数のコア上で、同一のタスクが異なるデータを処理する。既存のスケジューリングに関する研究の多くは、タスク並列性のみを対象としている。言い換えると、各タスクは 1 個のコア上で実行されることを想定している。しかし、多くの科学計算プログラムやメディア処理プログラムは、タスク並列性とデータ並列性の両者を有している。タスク並列性とデータ並列性の両者を活用することにより、どちらか一方のみを活用するよりも、高い性能を実現することが可能となる。本論文では、タスク並列性とデータ並列性の両方を活用するタスクスケジューリング問題（以下、タスク/データ並列タスクスケジューリング問題と呼ぶ）に対して、種々のアルゴリズムを提案する。

一般的にタスクスケジューリングは NP 困難な問題であり、現在までに数多くのヒューリスティックまたはメタヒューリスティックなアルゴリズムが提案されている。これらのアルゴリズムは実用的な時間内で近似解を求めることが可能である。リストスケジューリングは、最も有名なヒューリスティックアルゴリズムの一種である。各タスクに優先度を与え、優先度に従って、早い時刻から順にタスクをスケジュールする。本論文では、タスク並列性のみを考慮した既存のリストスケジューリングアルゴリズム拡張し、タスク/データ並列タスクスケジューリング問題に対するアルゴリズムを開発した。開発したアルゴリズムは 6 種類であり、優先度決定法がそれぞれ異なる。市販の数理最適化ソフトウェアを用いて解を求める方法と比較して、短時間で高品質の解が得られることを示した。

提案した 6 種類のアルゴリズムは、各タスクに対してひとつの優先度を与える。しかし、特定のひとつの優先度では、すべての問題に対して良い解を得ることが困難であることが判明した。そこで、状況に応じて 2 種類の優先度を切り替える、デュアルモードリストスケジューリングアルゴリズムを開発した。実験により、デュアルモードリストスケジューリングアルゴリズムは、1 種類の優先度しか持たない従来のリストスケジューリングアルゴリズムよりも、平均 2%、最大 10% 良い解を得ることを示した。

リストスケジューリングは非常に高速であり、かつ、実装が容易であるという利点がある一方、局所最適解に陥りやすく、問題によって得られる解の品質のばらつきが大きいという問題がある。そこで、本研究では、タスク/データ並列タスクスケジューリング問題に対する遺伝的アルゴリズムを開発した。遺伝的アルゴリズムはメタヒューリスティックアルゴリズムの一種であり、選択、交叉、突然変異という基本操作を繰り返すことにより、広範に解を探索する。本論文では、タスク/データ並列タスクスケジューリング問題に対する新たな

な染色体表現を定義し、その染色体定義に基づき、選択、交叉、突然変異の操作を開発した。開発した遺伝的アルゴリズムの逐次な実装に加えて、OpenMP による並列化実装も提案した。実験の結果、開発した遺伝的アルゴリズムはリストスケジューリングと比較して平均 5%、最大 13% 良い解が得られた。

上述のすべてのスケジューリングアルゴリズムは近似解を求めるアルゴリズムであり、最適解が得られる保証はない。しかしながら、例えばスケジューリングアルゴリズムを評価する場合など、真に最適なスケジュールを求める必要がある場合がある。そこで本論文では、解の最適性を保証するスケジューリングアルゴリズムを開発した。開発したアルゴリズムは分枝限定法に基づいており、4 種類の枝刈り法により解空間を削減しながら、効率的に最適解を探索する。最大 100 タスクからなる 160 の問題を用いて実験を行った結果、135 の問題について、12 時間以内に最適解を発見することができた。最適解が求まらなかった場合でも、既存のヒューリスティックアルゴリズムと同等以上の解を得ることができた。

本論文では、タスク/データ並列タスクスケジューリング問題に対し、大別して 4 種類のアルゴリズムを提案した。提案した 4 種類のアルゴリズムは計算時間と解の品質がそれぞれ異なる。リストスケジューリングアルゴリズム、デュアルモードリストスケジューリングアルゴリズム、遺伝的アルゴリズム、分枝限定法アルゴリズムの順で高速であり、その逆順で解の品質が高い。許容される計算時間に応じて、4 種類のアルゴリズムの中から最適なアルゴリズムを採用することが可能となる。

Abstract of Doctoral Thesis

Title : Scheduling Algorithms for Data-Parallel Tasks on Multicore Architectures

Doctoral Program in Advanced Electrical, Electronic and Computer Systems
Graduate School of Science and Engineering
Ritsumeikan University

リュウ ヨウ

LIU Yang

To meet the increasing demand for computational performance, the number of cores in embedded processors as well as general-purpose processors, has rapidly grown in recent years. How to fully utilize such processors with a high degree of parallelism has now become a more important issue than ever.

In general, the forms of parallelisms can be classified into task-parallelism and data-parallelism. The task-parallelism is achieved by the concurrent execution of different tasks on multiple cores in parallel, and the data-parallelism focuses on executing the same task with different input data sets on multiple cores. Many of existing task scheduling algorithms only consider the task-parallelism. In other words, each task is executed on a single core. However, most scientific and media applications often combine the two kinds of parallelism, which means, multiple data-parallel tasks are executed in a task-parallel fashion. This mixed-parallel approach significantly increases the scalability of parallelism. Many studies have shown that exploiting both task- and data-parallelisms often yields better performance than pure data- or task-parallelism. This paper addresses the task scheduling problem which takes into account both task- and data-parallelisms.

In this thesis, we provide an extensive survey on existing task scheduling algorithms. Since the scheduling problem is NP-hard, there are a large number of heuristics and meta-heuristics which aim to find near-optimal results in a practical time. List scheduling is one of the most popular heuristics for task scheduling problems, which assigns a particular priority to tasks, and schedules these tasks by the assigned priorities. In our thesis, we extend the traditional priority strategy to task scheduling for data-parallel tasks. We propose six list scheduling algorithms with different strategy of priority assignment. The experimental results demonstrate the effectiveness of the proposed algorithms against a commercial mathematical programming solver.

We also find that a specific static priority is hard to be effective against all applications. Next, we extend the simple list scheduling to use two static priorities switched during task scheduling. In our experiments, we compare the proposed algorithm with traditional list scheduling algorithms. The experimental results show that the proposed algorithm yields shorter scheduling length, by 2% on average and up to 10%, than pure list scheduling with a single priority.

The advantages of list scheduling algorithms and their variants produce results in a very short time and are relatively simple to implement. However, their acquired scheduling results are often far from optimal ones. In recent years, many studies have turned to meta-heuristics to solve task scheduling problems. Meta-heuristics provide certain algorithmic frameworks to search the solution space and avoid local optimal results, which are effective ways to improve the quality of results. In this thesis, we present an introduction of several popular meta-heuristics for task scheduling. Furthermore, an efficient method based on a genetic algorithm (one kind of meta-heuristics) is proposed to solve the task scheduling problem which considers both task- and data-parallelism. Different from traditional genetic algorithms for task scheduling, we propose a novel representation for the chromosome of task scheduling and corresponding genetic operators, aiming to reduce the search space and improve the computing speed. In addition to the single-thread implementation, we parallelize our algorithm with OpenMP to speed up our algorithm. Our experiments show that the proposed genetic algorithm finds

near-optimal schedules and outperforms the previously discussed list scheduling algorithms by 5% on average and up to 13%.

Although the heuristic and meta-heuristic algorithms produce sub-optimal scheduling lengths in a reasonable time, it is still desirable to obtain optimal scheduling lengths in some cases, for example, to evaluate heuristic algorithms. This thesis proposes an exact algorithm to find optimal results. The proposed algorithm is based on depth-first branch-and-bound search. We present four rules to prune non-optimal branches. The experiments show that our algorithm could find best schedules in a practical time. In our experiments with up to 100 tasks, the proposed algorithm successfully finds optimal schedules for 135 test cases out of 160 within 12 hours. Even in the case where optimal schedules are not found within 12 hours, the proposed algorithm finds better schedules than state-of-the-art heuristic algorithms.

As mentioned above, this thesis proposes broadly four algorithms for task scheduling with both task- and data-parallelisms. The four algorithms feature different characteristics on computational complexity and quality of results, and system designers can employ the one which best satisfies their requirements on computational complexity and quality of results.