

審査の結果の要旨

氏 名 フィン ゴク アン

本論文は、「Analyzing Performance Differences of Task Parallel Runtime Systems based on Scheduling Delays (スケジューリング遅延に基づいたタスク並列ランタイムシステムの性能差の解析)」と題し、英文10章から構成される。

第1章「Introduction (序論)」では、本論文の背景、動機、および貢献の概要を述べている。そして、言語処理系が自動的に動的負荷分散を行うシステム---タスク並列処理系---においては、プログラマが容易にプログラムを並列化できること、処理系の性能によって台数効果(並列化による性能向上)が左右されるために、性能理解が難しくなる場合があることを指摘している。

第2章は「Work-Stealing Scheduling Strategy (ワークスチーリングスケジューリング戦略)」と題され、本研究の背景および基礎であるワークスチーリングアルゴリズムに基づく負荷分散、スケジューリングについて述べ、その性質について述べている。

第3章は「Task Parallel Programming Models and Runtime Systems (タスク並列モデルと実行時システム)」と題され、本論で比較対象の評価とするものを含め、8種類のタスク並列システムの実装について述べている。

第4章は「Performance Analysis and Visualization Tools (性能解析と可視化ツール)」と題され、性能解析のためにこれまでに提案されてきた着眼点、可視化ツールについて述べている。

第5章は「Analyzing Performance Differences based on Scheduling Delays (性能の差をスケジューリング遅延に基づき解析する)」と題され、本論文で提案する性能解析にあたっての着眼点(スケジューリング遅延)、およびその解析方法について述べている。そこでは計算時間・並列度の積が、work (有効な仕事時間)、delay (本来実行されるべきタスクが実行されないことで生ずる遅延時間)、no-work (実行できる仕事がないことで生ずる空き時間)の3つの部分に分解できること、no-workがさらに、アプリケーションの並列度のなさに起因するno-work-app、スケジューラの遅れによるno-work-schedに分解されることを述べ、時間・並列度の積をこれらの要素に分解するための解析方法について述べている。

第6章は「Recording and Visualizing Computation DAG Traces (計算DAGトレースを記録、可視化する)」と題され、計算の実行トレースを有向非循環グラフ(Directed Acyclic Graph)としてモデル化できること、実行トレースをDAGとして保存するアルゴ

リズム (DAG Recorder)、保存されたDAGを可視化するツール (DAGViz) について論じている。また、大きなトレースに対してツールをスケールさせる方式について述べている。具体的には、DAGを保存する際に並列実行の観点から重要でないサブグラフを縮約し、少ないデータ量で保存する方法、保存されたグラフを必要に応じて異なる詳細度で可視化する方法、粗い表示をする際にDAG全体を読み出すこと無く、少ないデータアクセスで行うための方式、などについて述べている。保存されたDAGから、実際の並列度や抽出可能な並列度を計算する方法についても述べている。さらにケーススタディとして、タスク並列のベンチマークである Barcelona OpenMP Task Suite (BOTS) に対して、DAG抽出をすることによるオーバーヘッドを評価し、いくつかのアプリケーションから抽出されたDAGを表示、説明している。

第7章は「Task-Parallelizing PARSEC Benchmarks (PARSECベンチマークをタスク並列化する)」と題され、タスク並列用の新しいベンチマークについて述べている。PARSECはPrinceton大学によって開発され、Pthread、OpenMP、TBBで記述されているが、本研究で評価の対象とするタスク並列プログラミングモデル向けにプログラムを修正しており、本章ではその詳細について述べている。

第8章は「Evaluation (評価)」と題され、本手法をBOTS及びTP-PARSECの、合計21のベンチマークと5つの処理系の組み合わせに対して適用し、詳細な評価を行っている。評価にはマルチコア環境およびメニーコア環境の二つを用いている。

第9章「Related Work (関連研究)」では、本論文と関連する分野の既存研究について、性能分析、性能可視化、タスク並列のベンチマーク、の3つに分類して述べている。

第10章「Conclusion (結論)」では、本論文の提案骨子をまとめると共に得られた知見を要約している。

以上これを要するに、本論文では今後ますます重要になる、マルチコア、メニーコアプロセッサ上での並列プログラミングにおいて、実行時システムによる動的負荷分散を詳細に記録・保存する手法とツール、それを可視化する手法とツール、台数効果を詳細に解析し、アプリケーション由来の性能低下、実行時システム由来の性能低下などに分類して定量化する手法、を提案し、その有効性を詳細に評価しており、電子情報学の発展に寄与するところが少なくない。よって本論文は博士(情報理工学)の学位請求論文として合格と認められる。