

(続紙 1)

京都大学	博士 (情報学)	氏名	白 正陽
論文題目	Research on Parallel Hierarchical Matrix Construction (階層型行列生成の並列化に関する研究)		
(論文内容の要旨)			
<p>近年の計算科学分野では、境界要素分析や多体問題シミュレーションが多く行われている。これらのアプリケーションでは、物理要素間の相互関係を表す密行列を係数行列とする連立一次方程式を解くことが必要になることが多い。そのような密行列を直接扱ると、N個の物理要素に対する行列の要素数はN^2となり、大きなNに対しては必要なメモリ容量や計算時間が現実的に扱えないほど増大してしまう。この問題を解決するため、密行列の近似圧縮表現である階層型行列 (H行列)を用いることが提案されている。H行列は行列を階層的クラスタリングにより部分行列に区分けし、それぞれの部分行列をそれができる場合は2つの低ランク行列の積で近似表現したものである。H行列のデータ量は$O(N \log N)$となる。H行列を扱う計算の並列化・高速化に関する研究はこれまで多く行われているが、その構造の不規則さにより、CPUコア等の計算資源に計算を均等に割り当てるのが難しく負荷不均衡が生じてしまうという問題がある。</p> <p>本学位論文は、特にH行列の生成処理を対象として、上記の負荷不均衡の問題を、タスク並列言語を用いた動的負荷分散を採用することで解決することを提案し、その提案に基づく並列実装のスーパーコンピュータ上での性能評価を示すことで提案手法の有効性を主張するものである。</p> <p>本学位論文の第1章は序章となっており、研究目的と扱う問題、そして本学位論文の研究背景、主要な貢献と位置付けが述べられている。</p> <p>第2章では、第4章から第6章にかけて提案するH行列生成の並列実装の説明の準備として、H行列生成のアルゴリズムを説明している。具体的には、H行列の生成処理は部分行列への区分け決定処理と各区分けに対する行列要素を求めるフィル処理からなること、区分け決定処理はさらにCluster tree (CT) 構築とBlock cluster tree (BCT) 構築の処理からなることを述べたのち、それら各処理の詳細を説明している。</p> <p>第3章では、本学位論文で提案している並列実装に用いているタスク並列言語を紹介している。具体的には、並列言語Cilk PlusとTascellについて、それぞれのコード例を示したり言語機能・実装・性能特性を比較したりすることで説明している。</p> <p>第4章では、区分け決定処理の共有メモリ環境での並列実装の詳細が述べられている。CT構築処理とBCT構築処理のそれぞれについて提案する並列実装手法を説明している。特にCT構築においては、再帰ステップ間だけではなく再帰ステップ内の計算も並列化する二段階の並列化を行うという工夫を行うことで高い並列性を確保することを提案している。京都大学のスーパーコンピュータシステムLaurel2の単一ノード (18コアを持つCPUを2つ搭載) を用いた性能評価の結果を示し、提案手法により</p>			

高い並列性能が得られることを示している。

第5章では、第4章で述べられた並列実装を分散メモリ環境に拡張する2つの手法が述べられている。1つ目の手法であるDistributed Cluster Tree Construction(DCTC)では、CT構築を全計算ノードの全ワーカを用いて並列化する。また、この分散並列処理におけるノード間の通信量を小さく抑えるため、プログラマが指定する並列ループに対しては外部ノードのワーカからのスティール要求を拒否できるようにTascellを言語拡張することも提案している。2つ目の手法であるRedundant Cluster Tree Construction (RCTC)では、CT構築を全てのノードで冗長に行う。これら2つの手法による実装について、Laurel2のうち最大8ノードを用いた性能評価の結果を示し、DCTCではCT構築自体の性能向上は得られたものの、総合的な性能としてはRCTCのほうが良好な性能が得られたと結論付けている。

第6章では、第5章で述べた並列実装にフィル処理を結合した、H行列生成処理全体の並列実装を示している。特に、各部分行列に対するフィル処理の割り当てにも動的負荷分散を適用することで、負荷不均衡の問題を解消することを提案している。静的タスク割り当てを採用した既存実装との性能比較を九州大学のスーパーコンピュータITOの最大16ノードを用いて行うことにより、提案手法の有効性を示している。

第7章では、本研究の関連研究について紹介されている。

第8章は、学位論文全体の結論となっている。これまでに得られた研究成果の概要のまとめが述べられるとともに、今後の研究に関する展望について議論されている。

(論文審査の結果の要旨)

計算科学分野では、 $N \times N$ の密行列を扱う計算が多くあり、 N が大きくなると必要メモリ容量や計算時間が増大してしまう。この問題に対して密行列の近似圧縮表現である階層型行列 (H行列) が用いられている。H行列は、行列を階層的クラスタリングにより部分行列に区分けし、それぞれの部分行列を2つの低ランク行列の積として近似することにより表現したものであり、データ量は $O(N \log N)$ に削減される。一方で、このH行列を生成するための計算コストがかかる。本論文では、この生成処理に対して共有および分散メモリ環境向けの並列実装をタスク並列言語により行っている。具体的には、以下に示す研究成果を得ている。

(1)H行列生成における区分け決定処理の並列化の実現

H行列生成処理は、区分け決定処理とフィル処理という2段階の処理に分けられるが、本論文ではまず、区分け決定処理の並列化を検討している。区分け決定処理の計算は、形状が事前予測不可能かつアンバランスな木の構築をとまなうことに加え、構築における木の各ノードにおける生成コストも大きいという難しさを持ち、Cilk PlusやTascellなどのタスク並列言語を使って率直にプログラムを並列化しただけでは十分な性能が得られない。本論文では「二段階並列化」と呼ぶ並列化手法を提案・適用することでこの問題を解決し、共有メモリ環境において十分な性能が得られることを実験的に確認している。

(2)H行列生成における区分け決定処理の分散環境における並列化の実現

(1)で提案した共有メモリ環境での区分け決定処理手法を、本論文ではさらに分散メモリ環境に拡張している。Tascellは分散環境にも対応した動的負荷分散機能を提供するが、区分け決定処理はノード間並列化が適切な計算と適切でない計算とが混在しているため、Tascellをそのまま利用するだけでは十分な性能を得ることができない。そこで、ノード間のワークスティーラを許すか許さないかをプログラマが指定できるようTascellを機能拡張し、本区分け決定処理に適用している。また、この拡張機能によらない実装方法も提案し、分散環境での性能向上が得られることを実験的に確認している。

(3)H行列生成におけるフィル処理のタスク並列言語による実装

(1)(2)では区分け決定処理の並列化を行っていたが、次に、フィル処理の並列実行環境における性能向上も図っている。従来実装のフィル処理では、各ワーカが担当する計算を、コストが均等になるよう見積もったうえで事前に割り当てていたが、この見積もりが不正確であることによる負荷不均衡が生じていた。そこで、タスク並列言語Tascellの機能を活用して動的負荷分散を行うことでこの問題を解消した。計算時間に関して、従来の実装より最大約2.1倍の性能改善が得られ

ることを実験的に確認している。

以上を要するに、本論文は、**H**行列生成における区分け決定処理の並列化の実現とフィル処理のタスク並列言語による実装を行い、有効な並列化性能を達成し、もって科学技術における高性能計算の可能性を切り拓いている。したがって、本論文は博士（情報学）の学位論文として価値あるものと認める。また、令和5年2月15日、論文内容とそれに関連した事項について試問を行った結果、合格と認めた。また、本論文のインターネットでの全文公表についても支障がないことを確認した。

要旨公開可能日： _____ 年 _____ 月 _____ 日以降