階層型共有メモリプログラミング言語 MpC による TSP 並列処理

小山 浩生† 緑川 博子‡

1. はじめに

筆者らは，メタプロセスモデルというローカリティに着目した階層型共有メモリプログラミングモデルを提案し，これに基づくポータブルな API として MpC 言語を開発してきた．MpC コンパイラは，MpC プログラムをクラスタ向けには SDSM プログラム[1]へ，共有メモリマシンに対しては Pthread へ変換して並列実行を行えるようにしている．すでに，MpC がコモディティクラスタや共有メモリマシンにおいて，UPC や OpenMP と比較し，性能上の優位性があることを示した[2]．今回，MpC 言語を用いて TSP(Traveling Salesman Problem)の並列化を行い，共有メモリマシンやクラスタにおける並列実行の性能や，複数のマシンにおける MpC コンパイラと MpC プログラムの移植性について確認した．

2. メタプロセスモデルと MpC

2.1 メタプロセスモデル

メタプロセスモデルとは，従来の共有メモリモデルに，NUMA マシンなどにも対応できるような各プロセスへの共有分散データの階層構造（スコープ）を取り入れたプログラミングモデルである．メタプロセスとは，1 つの応用を協力して並列処理する複数のプロセス全体を指し，ユーザーにとっての実行単位である．

2.2 MpC

MpC 言語は，メタプロセスモデルをクラスタや共有メモリマシン上で実現するために共有データ型 shared と共有データ分散マッピング指定子を新しく導入して，C 言語を拡張したものである．新しく導入した shared 型は C の記憶クラス指定子として組み込み，共有データの分散マッピングは柔軟性の高い様々な割り付けが可能で，コンパイル時よりも実行時に処理を行うことで柔軟性を高めている．特定のコンパイラや実装系を前提とせず，アーキテクチャによらず移植性が高いことが特徴である．

3. TSP の並列化の方針

TSP を解くには厳密解法と近似解法があるが，ここでは実用的な時間で解を得ることができる近似解法を対象とし，構築法と改善法を組み合わせて用いた．構築法（Saving 法）で初期ツアーを決め，改善法（2-Opt, Or-Opt, 3-Opt の順）で初期ツアーを局所探索アルゴリズムに基づき改善していく．この手順を 1 試行とし，複数試行の中から最良解を最終の近似解とする．ただし，今回は局所探索範囲（交換候補の）制限を行って高速化を図るヒューリスティックは取り入れていない．TSP の並列化手法としては手始めに以下の基本的な２つを実装した．

3.1 複数回の試行の並列実行 （Trial 型）

近似解法では，初期ツアーを変えて複数の試行を行い，最もよい解を採用する．構築法での初期ツアー生成には，基点都市の決定に乱数を用いているため，乱数 seed を変えて複数の初期ツアーを生成し，並列に試行を行う．各試行の終了時に現在までの最良コストと比較し，改善されればロック(排他制御)して最良コストのツアーを更新する．また試行回数カウンターの更新も行う．したがって，処理全体のロック回数は試行回数 x2 と等しくなる

3.2 繰り返し構造の並列実行 (Parfor 型)

改善法における交換候補生成の繰り返しを並列化し，各候補採用時のコスト評価を並列に行う．コストが改善される場合のみロックをかけて最良コストとツアーを更新し，これ以上の改善ができなくなる（λ-optimal）までこれを繰り返す．そのためロックの回数は最良コストの更新回数とほぼ一致する．これを指定試行数分，繰り返す．

4. 実験結果

4.1 実行環境

用いた問題は TSPLIB[3]にあるものを対象にしたが，ここでは 1002 都市の pr1002 について報告する．表 1 のような 2 種の共有メモリマシンと PC クラス

† 成蹊大学大学院　工学研究科
‡ 成蹊大学　理工学部

タ（Giga Ethernet）で，試行数 16 としプロセス数 1～4(8)で並列実行し，実行時間と解の質を調べた．

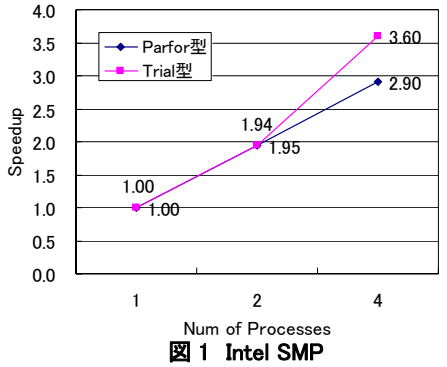

図 1　Intel SMP

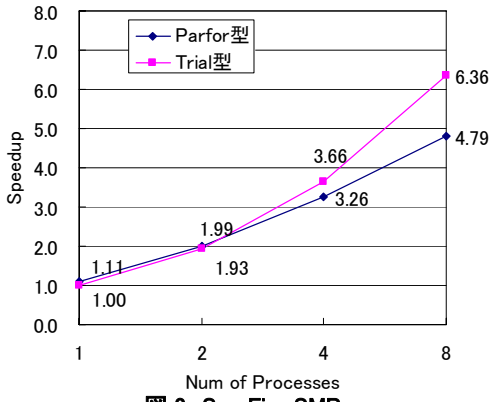

図 2　Sun Fire SMP

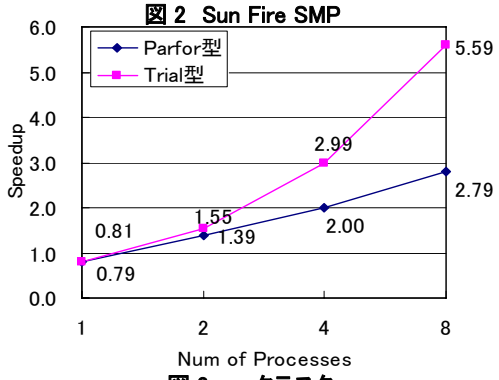

図 3　pc クラスタ

表1　実行環境

	Intel SMP	Sun Fire V40Z　SMP	PC クラスタ
CPU	Intel Xeon E5310 1.6GHz	AMD デュアルコア Opteron880 2.4GHz	Intel Pentium 4 3.00GHz
CPU 数	Quad コア × 2 チップ	Dual コア x 2 チップ	1CPU／ノード
OS	Linux FC5 kernel 2.6.19	SunOS 10	Linux kernel 2.6.9
メモリ	8GB	16GB	2GB／ノード

表 2　逐次処理の実行時間

	Intel SMP	Sun Fire SMP	PC クラスタ
実行時間[sec]	790.89	1224.48	1095.62

4.2　実行結果

得られた解の質は，Parfor 型，Trial 型共に TSPLIB に含まれる厳密解の約 3%～4%増の近似解が得られ，逐次処理による近似解と比べ，並列化による大きな変化は見られなかった．Trial 型は近傍探索・更新が逐次処理と同じ順序であるため，同じ近似解が得られる．一方，Parfor 型はツアー更新を並列に行い，逐次処理とは異なる順序で探索を進めるため，逐次処理の解とは必ずしも同じにはならない．

図 1, 2, 3 は，表 1 の各マシンにおける並列処理の速度向上比を示す．表 2 の各マシンでの逐次プログラム実行時間を基準としている．図 1, 2 の共有メモリマシンでは，Intel Quad コアマシンが SunFire に比べ，単体実行時間も高速で，速度向上比も優れている．いずれもロック数の少ない Trial 型が Parfor 型の性能を 1.2～1.3 倍ほど上回っている．

図 3 のクラスタの Parfor 型は，CPU 数を増やしても速度向上が低く，ロック回数の差（Parfor 型 32 回，Trial 型 1100～2600 回）がネットワーク通信オーバーヘッドとして現れている．またクラスタ単体ノードの逐次実行性能が高いため，全体として速度向上比が低い傾向にある．

5.　終わりに

MpC 言語を用いて，プログラムを書き換えることなく，クラスタ，共有メモリマシンの両方でコンパイル・実行が出来ることを確認した．現在の Parfor 型の 1 イテレーション毎の並列方式をブロック化することでロック数を減らすこと可能で，この効果も今後調べる予定である．また，ヒューリスティックを取り入れた探索アルゴリズムや，不規則な並列構造を含むプログラムに対する MpC の記述性についても，さらに評価を行う予定である．

参　考　文　献

[1] 緑川，飯塚："ユーザーレベル・ソフトウエア分散共有メモリ SMS の設計と実装"，情処論文誌 HPC,Vol.42, No.SIG9(HPS 3), pp.170-190　(2001)

[2] 緑川，飯塚："メタプロセスモデルに基づくポータブルな並列プログラミングインターフェース MpC",情処論文誌：ACS, Vol.46 No.SIG4(ACS9), pp.69-85,（2005)

[3] TSPLIB http://elib.zib.de/pub/Packages/mpestdata/tsp/tsplib/tsplib.html