TSUBAME 共同利用 平成 24 年度 学術利用 成果報告書

利用課題名 ポストペタ時代の大規模並列数値計算のための技術開発

英文：Research and Development on Large Scale Parallel Numerical Computations in the Post-Peta Era

須田 礼仁
Reiji Suda

東京大学 情報理工学系研究科
Graduate School of Information Science and Technology, the University of Tokyo
URL: http://sudalab.is.s.u-tokyo.ac.jp/~reiji/

邦文抄録（300 字程度）
本研究では，ポストペタスケールの計算機に向けて手法の開発を行っている．本稿では巡回セールスマン問題（TSP）について報告する．TSP はＮ都市をすべてまわる最短経路を求める有名な問題で，NP 困難である．我々は TSP を GPU で効率的に解くため，スレッド間通信を利用した反復局所探索の並列化手法を開発した．本手法で最大 2048 CPU，768 GPU まで良好な高速化率が確認された．

英文抄録（100 words 程度）
This project develops computational methods for post-peta scale machines. We report parallel solution of Travelling Salesman Problem (TSP). TSP is a classical problem in computer science. For a given number of cities N, find the shortest path that visits all N cities exactly once. This problem is classified as NP-hard. We analyzed an effective way of parallelizing Iterative Local Search using inter-thread communication. We observed good speedups using our communication technique for up to 2048 CPUs and 768 GPUs.

Keywords: ５つ程度
TSP, Optimization, GPU, MPI, Parallel

背景と目的

本研究では，巡回セールスマン問題（TSP）を取り上げる．TSP は情報科学でよく知られた組み合わせ最適化問題であり，定義が簡単で，既存研究も非常に多い．このため新しいアルゴリズムを実装した時に既存の研究成果と性能を比較することが容易であるという利点がある．

TSP は次のような問題である．N 個の都市と都市間の距離が与えられた時に，N 都市をちょうど 1 度ずつ回る経路で経路長が最小のものを求めるというものである． TSP は NP 困難な問題クラスに含まれており，純粋に全探索すると都市数に対して指数関数的な時間が必要である．

TSP にはいくつかの変種がある．対称 TSP では，都市間の距離が進む方向によらず同じである．従って無向グラフで表現できる．この対称性を用いることで，解の候補の数は半分になる．非対称 TSP では進む向きで距離が異なる問題である．

TSP を網羅的探索で解くには，N 都市をまわる経路をすべて数え上げることが必要となる．最初の都市は任意に決められるので，第 2 の都市は N - 1 都市のうちの一つとなる．第 3 の都市は N - 2 都市のうちのひとつとなるから，(N-1)! 通りの経路が存在する．これが非対称 TSP の解候補数である．対称 TSP では (N-1)!/2 となる．本研究では対称 TSP を取り上げる．

TSP はさまざまな応用問題があり，広く研究されている．多項式的な時間でそれなりによい近似解が得られる，発見的解法（ヒューリスティックス）が大量に知られている．我々の解法は simulated annealing を基礎としたものであり，探索時間が無限大の極限において大域的最適解が必ず見つかることを保証するアルゴリズムとなっている．

対称 TSP の発見的アルゴリズムに広く用いられている手法に，図1に示す 2-opt exchange と呼ばれる基本ステップがある．我々の手法も 2-opt exchange を用いており，これに反復局所探索（ILS）を組み合わせることで近似解を得る．我々の手法では ILS の所要時間の 90%以上が 2-opt exchange に消費されており，か

つ問題サイズが大きくなるとその割合も高まってゆく．これは，1 回の最適な 2-opt exchange を決定するのに $O(N^2)$ の時間がかかるためで，より簡単な摂動的手法であれば O(N) でもできるところである．しかしながら最適な 2-opt exchange は TSP にとって極めて有効であり，これを高速化することが求められる．

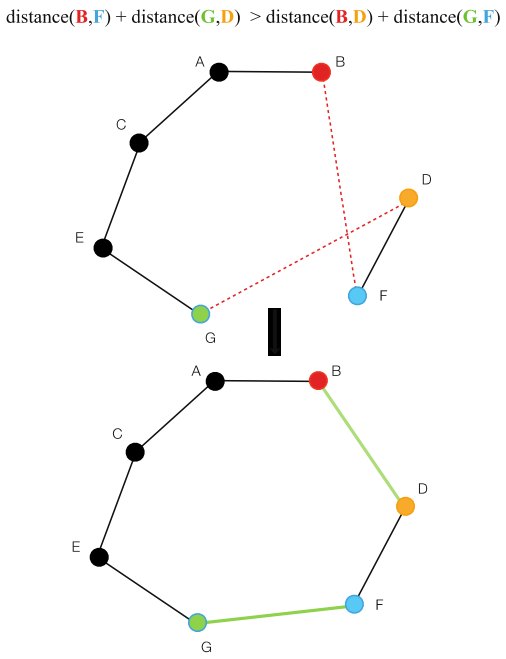

図1．2-opt exchange

近年 GPU を汎用計算に用いることが広く行われるようになってきた．とりわけ，高い並列性が少ない同期で実現できるような計算であれば，グラフィックスと関係なくても高速に実行できる．さらに，スーパーコンピュータも低消費電力高性能演算加速器として GPU を広く用いるようになっている．従って，多様なアプリケーションのために CUDA や OpenCL といった GPU プログラミング言語を用いる手法を開発することは極めて重要である．

概要

我々の手法は最適な 2-opt exchange を用いた反復局所探索に基づく．本稿では GPU を用いた実装の性能を評価する．

我々のアルゴリズムは，初期値 s0 としてランダムな経路を用いる．また double-bridge move を perturbation technique として用いる．

ノードをまたがる並列化としては，図 2 に示すような通信を間欠的に行うアルゴリズムを実装している．提案手法では，複数のノードが独立に最適化を行うが，時々図2に示すような通信を行い，それまでに見つかった最適解の情報を交換する．これにより，一つのノードがよい近似解を見つけたとき，その解の周辺を詳しく探索することができる．

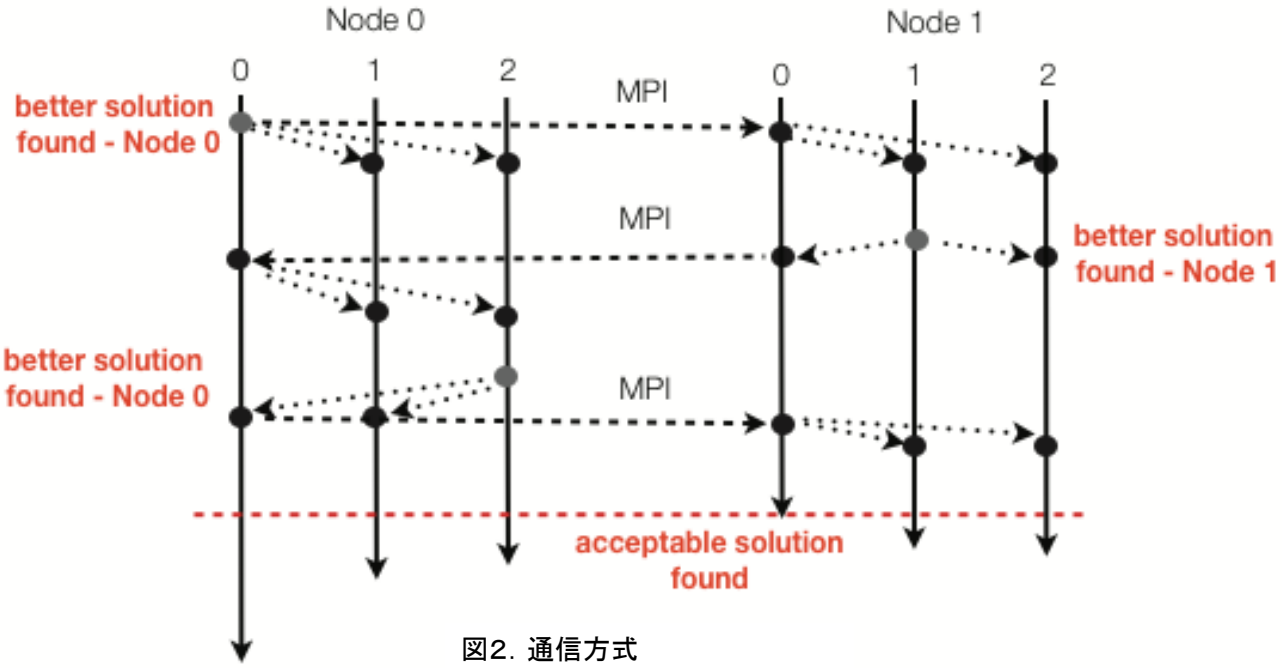

図2．通信方式

結果および考察

図3は GPU を用いた反復局所探索の性能を単一 CPU の性能との性能比で示したものである. 問題サイズが大きいほど, GPU による性能向上は高い. 逆に問題サイズが 200 より小さいと, 提案手法は CPU に比べてあまり高速でないが, これは GPU 版はデータ転送等の遅延の影響を強く受けるためである. 6 コア CPU を用いた CPU 向け並列実装と比較すると, GPU 実装は 5 倍から 45 倍ほど高速である.

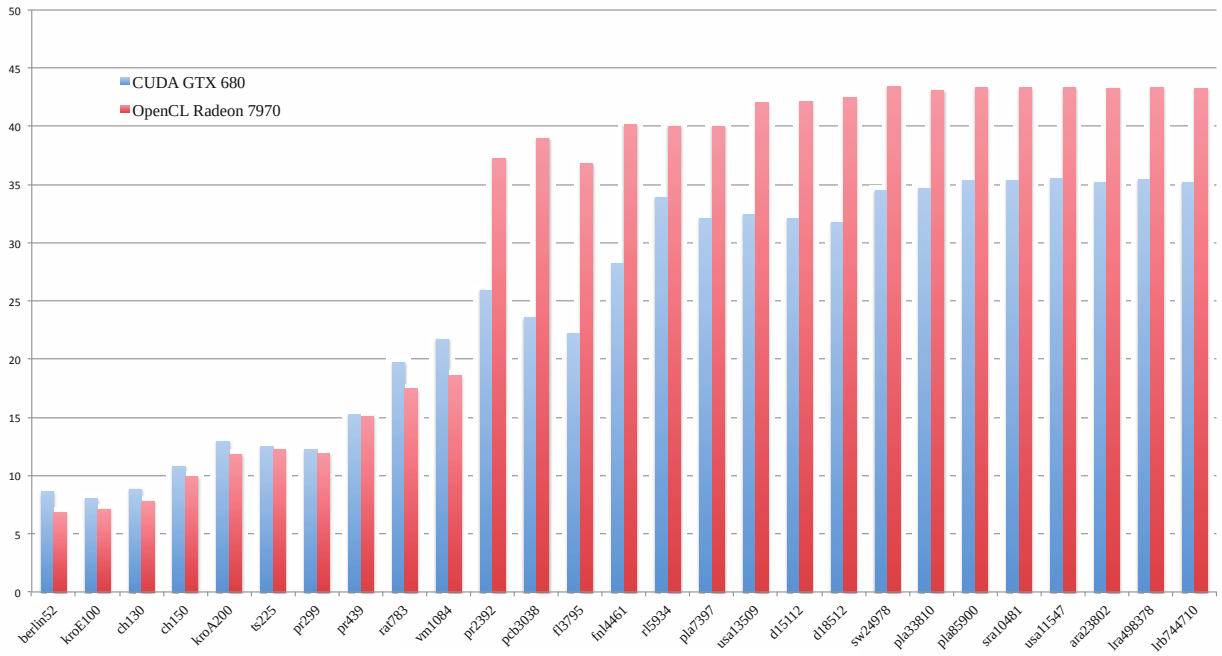

図3. 単一 CPU に対する高速化率

まとめ、今後の課題

本稿では, TSP の発見的解法の GPU 実装について報告した.

今後の課題としては, より複雑な局所探索である 2.5-opt, 3-opt, Lin-Kernighan などを実装することがある. これらの手法の並列実装はさらに複雑であり, 努力が必要である. また, 近傍の枝刈りなどの手法により, 低コストで解の質を改善することが考えられる.