

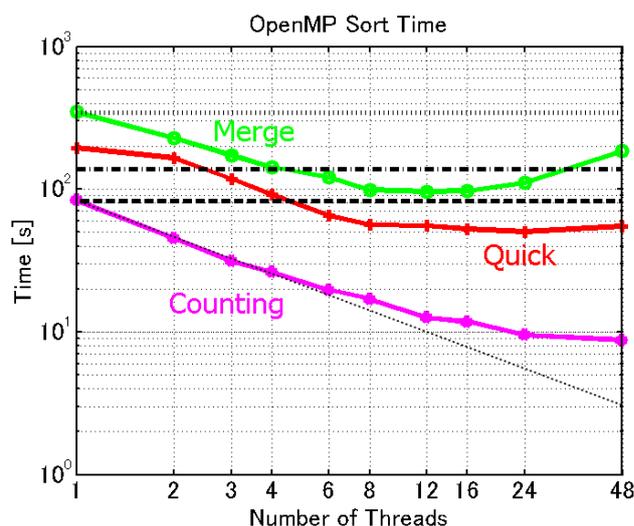
OpenMP を用いたソーティングのスレッド並列化

梅田 隆行 大矢 宗平
名古屋大学太陽地球環境研究所

近年の HPC 用計算機に用いられる CPU は、スカラ型マルチコア CPU が主流である。現在の PC クラスタ型汎用スーパーコンピュータでは、ノードあたり 2-4 個の CPU が搭載でき、ノード当たりのコア数は 8-64 個が主流となっている。一方で、ソーティングアルゴリズムには様々な用途があるものの、並列化された（フリーな）サンプルプログラムはほとんど出回っていない。本研究の目的は、OpenMP を用いて簡単な並列化を行った Fortran 版ソーティングアルゴリズムの性能評価を行うことである。

本研究では、単精度実数配列 $a(n)$ をその整数部 $\text{floor}(a)$ に対して昇順になるように並べ替えるソーティングアルゴリズムの中で、高速かつ容易に並列化可能なクイックソート、マージソート、カウンティング（計数）ソートの 3 つの性能評価を行った。クイックソートとマージソートは `omp task` ディレクティブを用いた再起アルゴリズムを採用した。カウンティングソートは、分布の数え上げとデータの並び変えを `omp do` ディレクティブにより並列化した。データ数は $n=1,200,000,000$ 、要素数は $m=1,000$ とし、強いスケーリングのテストを行った。速度計測には、名古屋大学太陽地球環境研究所のスーパーコンピュータ DELL PowerEdge R815 (Opteron 6174 [12 コア・2.2GHz]×4CPU, 96GB メモリ/ノード) を用いた。

再起アルゴリズムを `omp task` ディレクティブにより並列化した場合、task 起動のオーバーヘッドが大きいため計算速度が低下することが分かった。また task が不均一に起動されることにより、使用コア数が増加するとスケーラビリティが著しく低下することが分かった。これらは、今後の OpenMP のバージョンアップにより改善されるものと期待している。一方、カウンティングソートは、16 コア使用時にスケーラビリティが 50% を下回り、48 コア使用時には 20% 未満であった。効率が落ちる原因として、使用スレッド数 $p \times$ 要素数 m の累積分布を作成するときの計算負荷が増加している（本来ならば $O(m)$ ）ことが挙げられ、アルゴリズムの改善が必要である。



本研究は、科学研究費補助金 新学術領域研究 No.21200050 及び若手研究(B)No.23740367 の助成を受けて行われた。名古屋大学太陽地球環境研究所スーパーコンピュータ DELL PowerEdge R815 の計算リソースは、同研究所の計算機利用共同研究及び名古屋大学 HPC 計算科学連携研究プロジェクトにより提供された。