

1. 序論

情報処理の高速化かつ多様化に従い様々な分野で高速化が追及されている。高速化を行うためには複数のCPUに仕事を分担させて並列処理を行う並列計算機が必要とされている。しかし、一般的に、並列計算機は非常に効果であり、必ずしも必要な性能の並列計算機を使用できるとは限らない。そのため、複数の計算機をネットワーク接続することにより、仮想的に並列計算機を構成するクラスタ(Cluster)処理が注目されている。また、高速化が求められている分野の1つに画像処理がある。画像はデータが膨大であるため、画像の並列処理には高い関心が集められている。

本研究は、クラスタ処理を行うソフトウェアの1つであるMPI(Message Passing Interface)¹⁾を用いて、画像処理の高速化を目的とする。

2. 研究内容

本研究ではMPIを用いて、画素を対象とした並列画像処理を行う。近傍データに対する局所的な積和演算処理の代表として、画像からエッジを抽出する処理がある。本研究では、その中でもよく利用されるソベルオペレータのプログラム²⁾を並列化した。

エッジ抽出アルゴリズムは以下の式を計算することがアルゴリズムの中心になる。

$$f(i, j) = \sum f(i+m, j+n) * W(m, n)$$

ただし、 $f(i, j)$ は $n*n$ サイズの画像、 $W(m, n)$ は $m*m$ サイズのマスクである。また、 m は3~11程度の奇数値であり、本研究では $m=3$ としている。

本研究では、処理の対象画像を複数の部分に分割し、各部分の画像の処理各CPUに割り当てることにより、並列化を行った。

以下では、 $n*n$ 画像を4台のCPUを用いて処理を行う場合について述べる。

各プロセッサは、0~3の識別番号(rank)が割り当てられている。このとき、 $n*n$ 画像 (i, j) ($0 \leq i, j < n$)はサイズ $n*(n/4)$ の部分画像 (i, j) ($0 \leq i < n, nr/4 \leq j < n(r+1)/4$)に分割され、エッジ検出が行われる。ただし、 r はCPUの識別番号(rank)である。

以上を用いてプログラムし、画像の処理計算にかかる時間をCPU数別に計測を行う。

3. 結果・考察

図1に、画像ごとのエッジ検出にかかる処理時間とCPU数の関係を示す。CPU数を1から2に増やした

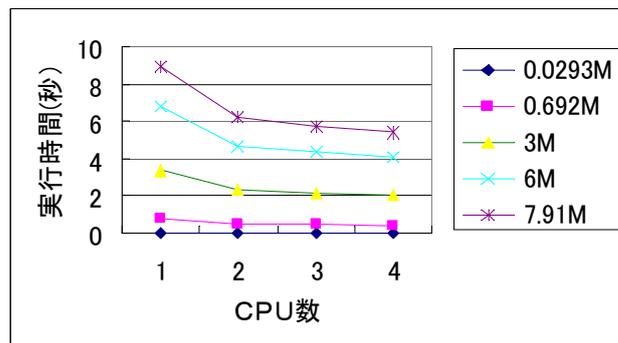


図1 処理時間とCPU数の関係

時は、実行時間が大きく変わったが(本研究では約2/3の短縮)、3~4と増やしていった時は、大きな変化は見られず、実行時間はCPUの数を増やすにつれ収束していった。また、容量が30kb程度の軽い画像では、CPU数が4つの場合に、1台で実行したときよりも遅くなるのがしばしば起きた。これは通信・同期にかかる時間が全体の処理時間の大きな割合を占めていたからだと考えられる。

画像の容量や計算機のスペックによって、処理速度や通信・同期にかかる時間が変わってくるが、処理範囲を分割して画像を並列処理するのは、計測結果を見る限り、CPU数が2つ、多くて3つが実用的だと思われる。

4. 結論

本研究では、MPIを用いて画像処理の1つであるエッジ抽出を行い、その時間を計測した。サイズの大きな画像の場合はMPIを用いて並列化を行うことにより、時間の短縮が得られる。一方、サイズが小さい画像の場合は、1台のときよりも遅くなることもあり、MPIの使用が必ずしも効率的だとは言えない。従って、MPIを使用する際には、使用できるCPU、ネットワークの性能やその他の計算機環境に応じてプログラミングする必要がある。

参考文献

- 1) 渡邊 真也:MPIによるプログラミングの基礎,
<http://mikilab.doshisha.ac.jp/dia/smpp/cluster2000/PDF/chafter02.pdf>
- 2) 美濃 道彦:並列画像処理, コロナ社(1999)