

1. 序論

何事においても仕事の処理時間は短い方がよく、計算機の処理は常に高速なものが求められている。高速化の手段の1つとして複数のプロセッサを用いることによる並列処理があげられる。しかし、複数のプロセッサを持つ並列計算機は非常に高価であるために容易に利用することができない。そこで複数の計算機をネットワークで繋ぎ仮想的並列計算機とすることで安価で並列処理が可能となる。

本研究ではMPI(Message Passing Interface)を用いたMPICH2というソフトウェアを使用しwavファイルからmp3ファイルへの変換においてどれ程の時間短縮ができたのかを検証する。

2. 研究内容

MPIとは並列計算を利用するための標準化された規格である。複数のCPUが情報をバイト列からなるメッセージとして送受信することで協調動作を行えるようにする。自由に使用できる実装としてはMPICHなどがある。ライブラリレベルでの並列化であるため、言語を問わず利用でき、プログラマが細密なチューニングを行えるというメリットがある。

- 1) 本研究では性能の異なった4台のPCに性能の高いものから順に、予め分割しておいたwavファイルをサイズの大きなものから振り分けmp3ファイルに変換し、別に用意した性能の良い1台のPCで分割する前のファイルの変換を行った場合との処理時間の違いを検証する。
- 2) 次に4台のPCに均等のサイズのファイルを振り分け変換を行った場合と、先ほどのスペックごとにファイルサイズを変更して分割し変換を行った場合の処理時間の違いを検証する。

変換を行う際に予め分割したwavファイルのサイズは約100MBと300MBと150MBである。研究に使用したPCの詳細は表1にまとめる。

表1. 開発環境

ホスト名	使用したホスト				
	Murayoshi	Hokazono	Kanehisa	Magician2	FM
OS	Windows XP Pro	Windows XP Pro	Windows XP Pro	Windows XP Pro	Windows XP Pro
CPU	Pentium 1.60GHz	Pentium 1.60GHz	Pentium 1.60GHz	Pentium4 2.50GHz	Pentium4 3.20GHz
RAM	512MB	512MB	512MB	0.99GB	1.99GB
振分けたファイル	100MB	100MB	100MB	300MB	600MB
	150MB	150MB	150MB	150MB	×

3. 結果・考察

表2. 処理時間

ファイルサイズ	ホスト数	PC名				
		Murayoshi	Hokazono	Kanehisa	Magician2	FM
600MB	1	197.16	85.87	102.42	53.94	51.89
100MBと300MB	4	18.12				×
150MB	4	20.02				×

表2に今回の処理の実行結果を示す。処理時間の決定には誤差の無いように全ての処理をそれぞれ10回ずつ行いその平均の数値を採用している。

Murayoshi, Hokazono, Kanehisa, Magician2はそれぞれホスト1台で600MBのファイルを変換した結果と、4台で分割したファイルを処理した結果を表しており、FMは1台で変換した結果を表している。

低スペックのPC4台でそれぞれに適したサイズのファイルを振り分けて変換を行った方が高スペックのPC1台で変換を行うよりも短い処理時間で変換できており、低スペックのPCに均等のサイズのファイルを振り分けて変換を行ったものと比較しても同様の結果を得ることができた。

4. 結論

Wavファイルをmp3ファイルに変換する場合は並列処理を行うことで処理時間を短縮することが可能である事がわかった。さらに、高スペックのPCと低スペックのPCでそれぞれ処理するファイルのサイズを変更した方が更なる処理時間の短縮に繋がる事が、そして低スペックのPCでも並列処理を行えば高スペックのPC1台の処理速度よりも早くなる事がわかった。

本研究ではファイルを分割してそれぞれのPCに適したサイズのファイルを振り分ける動作と、処理後に圧縮されたファイルを結合する動作の処理時間を含めていない為、この点を考慮して実験を行うことが今後の課題である。

参考文献

- 1) MPICH2
<http://www.mcs.anl.gov/research/projects/mpich2/>
- 2) 午後のこ〜だ,
http://www.marinecat.net/free/windows/mct_free.htm
- 3) P.パチェコ 著, 秋葉博 訳: MPI 並列プログラミング, 培風館 (2001)