並列化コンパイラのためのデータ自動分割法の検討

指導教官：市川 周一　　学籍番号：931714　藤村 佳克

1 HPF とデータの自動分割

並列化言語の一つの HPF(High Performance Fortran) では，directive で並列計算機上へのデータ分割や配置が記述可能である [1]. HPF ではユーザの意志で並列性を制御できるが，より容易に並列性を利用するには自動的にデータを分割配置することが望ましい．本研究では，並列化コンパイラのためのデータの自動分割法を検討する．

本研究の最終目的は並列処理により実行時間を短縮することである．実行時間を短縮するにはデータ分割により計算負荷を分散して計算時間を短縮すれば良いが，データ分割に伴って通信時間が発生し実行時間短縮の妨げとなる．従ってデータ分割にあたっては，通信量の低減や通信と計算の重畳も考慮して実行時間の最適化を図る必要がある．

一般に計算時間や通信時間は実装に依存するため，最適なデータ分割は実装に依存して決まる．以下では実装依存の要素をパラメータとして計算システムをモデル化し，単純な計算例を使って最適なデータ分割を求める方法を示す．

2 計算モデル

計算例として二次元画像処理の Low Pass Filter を用いる．Low Pass Filter は近隣のデータを使用して計算を行うため，データのブロック分割により通信をブロック辺縁部だけに抑えることができる．分割されたブロックは各プロセッサに配置され並列に処理される．偏微分方程式を差分解法で解く場合も同様の手法により計算が行われる．

各プロセッサは，ブロック辺縁部で使用するデータを近隣ブロックから受信し，近隣ブロックの計算のために辺縁部のデータを送信してから自分の担当する要素の計算を行う．この計算モデルをモデル１とする．外部からのデータに依存しているのはブロック辺縁部だけで，ブロックの内部は通信と関係なく計算を行うことができるので，通信と計算を重畳して遅延時間を隠すことができる．通信と計算が完全に重畳可能な理想的計算モデルをモデル２とする．

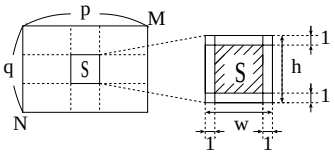


図 1: 配列のサブブロックへの等分割

配列は図１のようにプロセッサ n 個にブロック分割する．

$$n = pq$$

p は配列の幅の分割数，qは配列の高さの分割数である．図１の斜線部の周りが通信を必要とする辺縁部で幅は１である．配列ブロックの幅を M，高さを Nとすると，サブブロック Sの幅 wと高さ h は

$$w = \lceil \frac{M}{p} \rceil, h = \lceil \frac{N}{q} \rceil$$

となり，計算時間 T_aは

$$T_a = \alpha wh$$

で求まる．αは１つの要素の計算時間である．各サブブロックの受信量を S_r，送信量を S_sとすると，

$$S_r = (w+2)(h+2) - wh$$
$$S_s = wh - (w-2)(h-2)$$

となる．プロセッサ間ネットワークが理想的であると仮定すれば，通信時間 T_cは

$$T_c = \beta(S_r + S_s) + \gamma$$

で表わされる．βはデータの転送幅，γは遅延時間である．プロセッサ間ネットワークが理想的な場合はこの一次近似が成立するが，実際にはプロセッサ数の増加に応じて通信遅延が増大する場合がある．本研究では，この通信時間を以下の式でモデル化する．

$$T_c = \beta'(S_r + S_s) + \gamma' n$$

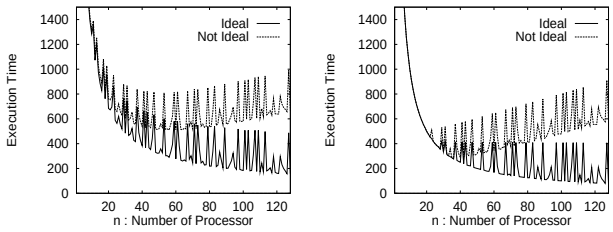
β'，γ'は台数を考慮した場合の転送幅，遅延時間である．このとき実行時間の見積りは以下のとおりである (T_1，T_2はそれぞれモデル１，モデル２の実行時間).

$$T_1 = T_a + T_c$$
$$T_2 = max(T_a, T_c)$$

以上の計算モデルを使用し T_1，T_2をネットワークの違いも考慮してシミュレーションする．

3 シミュレーション結果

計算モデルの各パラメータは実装依存であるが，本研究ではαを 0.01，βを 0.1，γを 4.0，β'を 0.1，γ'を 4.0 と設定した．使用する配列ブロックは $1000 * 1000$ の正方行列とする．以下，Ideal をプロセッサ間ネットワークが理想的な場合の結果，Not Ideal をプロセッサ数を考慮した場合の結果と表す．



(a) モデル１　　(b) モデル２

図 2: プロセッサ数と実行時間

図２ (b) の結果により，計算と通信の重畳によって実行時間が短縮されることが確認できた．プロセッサ数が素数であるとサブブロックの周長が大きくなるデータ分割しかできないため，通信量が増加して実行時間が増加している．Ideal ではプロセッサの増加に伴い計算量，通信量が減少し実行時間が短縮されるが，Not Ideal ではプロセッサ数を一定限度以上に増やしても通信時間の増加により性能は向上しない．

以上の結果より，以下の２点が確認できた．

・計算と通信の重畳は実行時間の短縮に有効である．

・過剰なプロセッサに負荷を分散すると実行時間が延びることがある．

4 まとめ

本研究により，最適なデータ分割が実行時間を短縮することが確認できたが，最適なデータ分割の決定は一種の組合せ最適化なので解くのは容易ではない．自動並列化コンパイラの PARADIGM[2] では凸計画法により最適化を行っているが，これでは近似解しか求まらない．今後の課題として真の最適解を求める方法を考えたい．

参考文献

[1] 中谷登志男：HPF コンパイラの現状と課題，電子情報通信学会論文誌 D-1，Vol.J78-D-1，No.2，pp.142-148，1995.

[2] M.Gupta，P.Banerjee：PARADIGM：A Compiler for Automatic Data Distribution on Multicomputers, in Proceedings of the 7th ACM International Conference on Supercomputing，pp.87-96，Tokyo，Japan，1993.