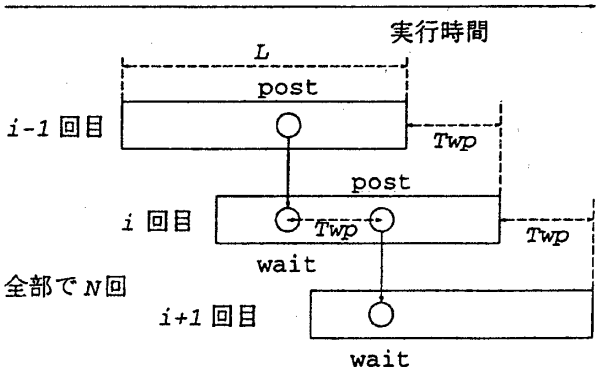
Doacross ループの実行時間と最適プロセッサ数

2E−1　児島 彰　藤野 清次　弘中 哲夫　高山 毅
広島市立大学 情報科学部 情報工学科

1 はじめに

ループを並列化するとき、ループ運搬依存がないときは、Doall ループとして並列化し、ループ運搬依存があるときは、同期操作がある Doacross ループとして並列化する。本研究では、Doacross ループの実行時間と、最速の実行に必要なプロセッサ数の見積りについて述べる。これまでに Cytron によって、Doacross ループの実行時間と最適プロセッサ数の見積りについての研究がなされている [1][2]。福田は、1重ループに対して、キャッシュなどの影響を考慮して、1回目の繰り返しと、2回目以降の繰り返しで実行時間が違うモデルでの研究を行っている [3]。従来の研究では、繰り返し間の依存による待ち時間の部分は考慮されているが、同期操作自体での消費時間が考慮されていなかった。また、実際の並列計算機での評価が行われていなかった。




```
Doacross ( i = 0 ; i < N ; i++ ) {
        { calc_part_1; }  .....(1)
        wait(i-1);
        { calc_part_2; }  .....(2)
        post(i);
        { calc_part_3; }  .....(3)
}
```

図 1: Doacross の構造

2 Doacross ループの解析

本論文で対象とする Doacross ループは、1重 Doacross ループで、繰り返しの実行時間が一定のものとする。これは、中小規模のループで条件分岐など各繰り返しの実行時間を不均一にする要素がループ内にないものが対象になると考えられる。

図1では、基本的な Doacross の構造を示している。i 回目の繰り返しでは実行を行う前に、$i-1$ 回目の演算部 (2) の実行を待つ必要があり、post−wait の同期操作を行っている。Cytron の研究と同様に同期操作自体には時間がかからないとする理想的な状況を仮定すると、依存関係の解決時間のみで Doacross の実行が決定される。L を1回の繰り返し時間、N を繰り返し回数、P を最大プロセッサ数、T_{wp} を post−wait 間での演算時間とすると、繰り返し間の遅延は、$\frac{L}{P} < T_{wp}$ のとき T_{wp}、$T_{wp} \leq \frac{L}{P}$ のとき $\frac{L}{P}$ となるので、実行時間は

$$\begin{cases} L+(N-1)\cdot T_{wp}, & \text{for } \frac{L}{P} < T_{wp} \\ L+(N-1)\cdot \frac{L}{P}, & \text{for } T_{wp} \leq \frac{L}{P} \end{cases}$$

となり、最速実行に必要最小なプロセッサ数は、$\lceil \frac{L}{T_{wp}} \rceil$ となる。同様に、i 回目の繰り返しが $i-j$ 回目の繰り返しに依存する場合、実行時間は、

$$\begin{cases} L+(N-1)\cdot \frac{T_{wp}}{j}, & \text{for } \frac{L}{P} < \frac{T_{wp}}{j} \\ L+(N-1)\cdot \frac{L}{P}, & \text{for } \frac{T_{wp}}{j} \leq \frac{L}{P} \end{cases}$$

となり、最速実行に必要最小なプロセッサ数は、$\lceil j\frac{L}{T_{wp}} \rceil$ となる。また、依存関係が複数ある場合は、待ち時間を最も長くする依存関係に全体の実行時間は支配され、その依存関係のための同期に対する上記の値が全体の実行時間、最適プロセッサ数となる。

3 並列計算機での実測

4 CPU 共有メモリ型並列計算機 DEC Alpha Server 2100 4/275 を使用し、Doacross ループの計測を行った。プログラムは OSF/1 Pthread で作成した。演算部分に

Estimation of the Time and the Optimal Number of Processors for Doacross Loops
KOJIMA Akira, FUJINO Seiji, HIRONAKA Tetsuo, TAKAYAMA Tsuyoshi
Department of Computer Science, Faculty of Information Sciences, Hiroshima City University
Hiroshima, 731-31, Japan

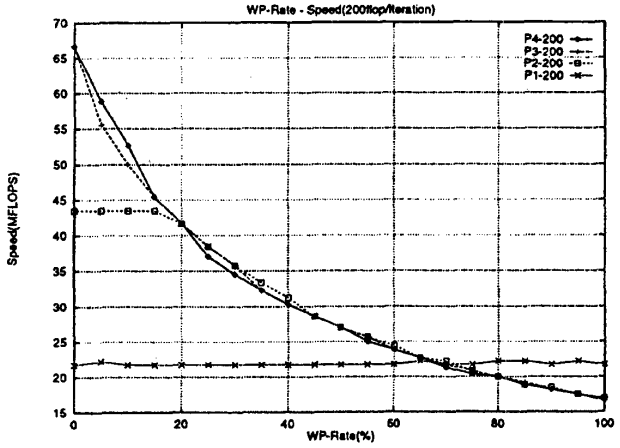


図 2: 浮動小数点演算 200 回/iteration

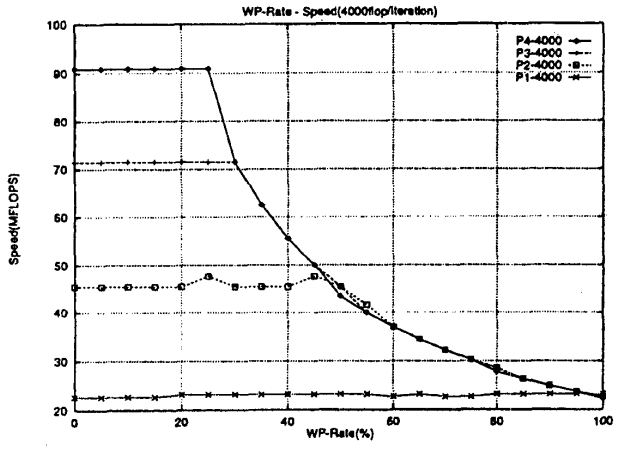


図 3: 浮動小数点演算 4000 回/iteration

は、浮動小数点の加算と乗算を 1:1 で行うコードで重み付けし、全体の演算量、post–wait 間の演算量を変化させ、全実行時間、プロセッサ数の関係を計測した。

計測結果のグラフを図 2 、 3 に示す。縦軸は速度 (MFLOPS)、横軸の WP は wait–post 間の演算量の全ループに対する比率 (%) を示す。グラフ中の P4-200 は、1 回の繰り返し中に浮動小数点演算が 200 回ある Doacross を、4 プロセッサで実行したことを示す。

全体の計算量が、かなり多いときは、先に述べた実行時間、最適プロセッサの式は、ほぼ成立していることがグラフから読みとれる。しかし、計算量が少ないときは、同期操作自体の時間の影響が大きく、先に述べた実行時間、最適プロセッサの式は、成立していない。

4 考察

最近のパイプライン化されたプロセッサでは、分岐のない部分の演算処理が高速なのに対して、分岐があるときには先読みのペナルティが大きなものとなっている。同期操作の wait 操作は、post 側から通知される通過条件成立フラグを待つループである。ここに現れるループ終了判定での分岐が先読みペナルティとなり、特に中小規模の Doacross ループでは無視できないものとなる。また、post 側からの通知が共有メモリ上の変数を介して行れるので、この部分でのキャッシュのミスヒットも時間のかかる要因となっている。

同期操作での実行時間を X とすると、繰り返し間の遅延は $Twp+X$ となるので、先に上げた並列化したの Doacross の実行時間は、$L+(Twp+X)\cdot(N-1)$ となる。1 プロセッサで実行したときのループの全実行時間は、$L\cdot N$ であるので、$L\cdot N < L+(Twp+X)\cdot(N-1)$、すなわち、$X > L-Twp$ のときは、1 プロセッサで実行したときの方が速いことがわかる。また、今回、計測したプログラムで用いた同期操作は、特定変数領域を while ループで待つという簡単なものであるが、この式とグラフから、浮動小数点演算の約 60 回分相当の時間、$3\mu sec.$ 程度かかっていることが分かる。

5 おわりに

本研究では、Doacross の実行時間と最適プロセッサ数を見積る議論を行った。実際の並列計算機で実測を行い、通常の中小規模程度のループでは、同期操作自体での消費時間の影響が無視できないことを確かめた。ループ内に条件分岐などがあり、繰り返しの実行時間が均一でない場合についての研究は今後の課題である。

参考文献

[1] Ron Cytron: "Doacross: Beyond Vectorization for Multiprocessors (Extended Abstract)", Proc. of Int. Conf. on Parallel Processing pp.836-844 (1986)

[2] Ron Cytron: "Limited Processor Scheduling of Doacross Loops", Proc. of Int. Conf. on Parallel Processing pp.226-234 (1987)

[3] 福田 晃: "イタレーション実行時間が異なる 1 重 Doacross ループの並列処理における最適プロセッサ数", 電子情報通信学会論文誌 D-I Vol.J75-D-I No.7 pp.450-458 (1992)