密結合マルチプロセッサ上の ＫＬ１並列処理系の評価

細井　聡、小沢　年弘、Mark Feldman、服部　彰
富士通株式会社

１．はじめに

我々は、第５世代プロジェクトの一環として、並列推論マシンＰＩＭの研究を進めている。今回、このＰＩＭ上の核言語であるＫＬ１の並列処理系を密結合マルチプロセッサ（Sequent社のBalance8000）上に試作した。そして、台数効果、ＣＰＵ占有率、負荷分散方式などについて測定、評価を行ったので報告する。

２．ＫＬ１並列処理系の構成

１つのＰＥを１つのUnixプロセスとして発生させ（ＰＥ数はパラメータにより可変）、それぞれを、ＫＬ１Ｂコードを解釈実行するエミュレータとして並列に実行させる。

また、共有メモリ上の共有資源の排他制御、および競合によるオーバーヘッドをできるだけ減らすようにしている。たとえば、ヒープ、フリーリスト（ゴールレコード）など、実行時に動的に確保する資源は、あらかじめＰＥごとに分配してある。

３．台数効果

各テストプログラムに対する台数効果を図１に示す。ＰＥ数が少ないうちはほぼＰＥ数に比例するが、ＰＥ数が多くなるにつれてややなまってくる。テストプログラムにより多少差があるが、台数効果は８ＰＥのとき７倍弱であった。

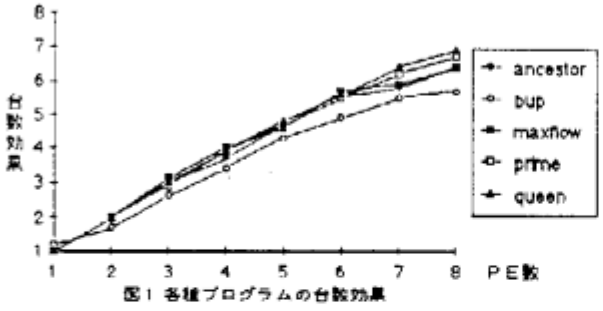

図１ 各種プログラムの台数効果

４．ＣＰＵ占有率

これは、エミュレータの各処理が全実行のおよそどのくらいを占めているかを表わす。実際には、各処理をそうはっきりと分けることはできないので、多少、誤差が含まれてしまっている。bupの場合のＣＰＵ占有率を図２に示す。

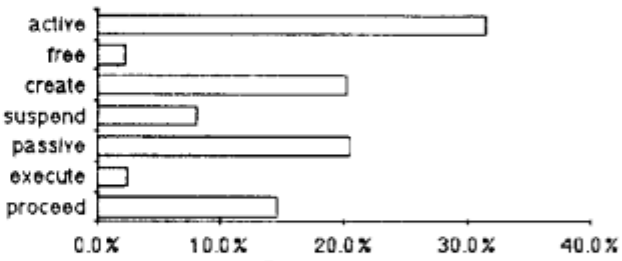

図２ ＣＰＵ占有率 (bup)

各処理の概要は以下の通り。

1)proceed　ゴールのデキュー（ＰＥアイドル状態、レディキュー競合状態を含む）とレジスタへのロード
2)execute　ＴＲＯ実行
3)passive　ガード部実行
4)suspend　サスペンド／リジューム
5)create　新ゴールの作成とエンキュー（引数準備命令を含む）
6)free　ゴールレコードをフリーリストにつなぐ
7)active　アクティブユニフィケーションおよび、create関連以外のレジスタ転送命令など

図２を一見すると、active,passive,createが大きな割合を占めているように見える。しかし、これらに対応する部分はＫＬ１Ｂ中間コードの数が多いため、命令を逐一フェッチして解釈実行するオーバーヘッドが目立ってきてしまっていることを考慮する必要がある。（これらの命令は、ネイティブコードに落とせばもっと軽くなるはずであり、したがって全体に占める割合も小さくなるであろう。）

一方、suspend と proceed は合わせて２０％強となっている。これは、ゴールをサスペンド／リジュームしたり、コンテクストスイッチ（ＣＳＷ）を起こしたりするコストが高いのに加えて、これらの頻度も大きいためであると思われる。（およそ5.5リダクションに１回サスペンドが起こっている。また、ＰＥアイドル状態や　レディキュー競合状態の割合が小さいことは後述する。）さらに、executeの割合が少ないのも、execute自体の処理が軽いというよりは、あまりＴＲＯで実行できないからだと言えるだろう。

Evaluation of KL1 on a sharedmemory multiprocessor
Akira HOSOI, Toshihiro OZAWA, Mark FELDMAN, Akira HATTORI
FUJITSU Ltd.

5．負荷分散について

1)まず、定義を明確にしておく．

「ＰＥがアイドルである」とは、「実行すべきゴールがなくてＰＥが空回りしている」状態のことをいう．そして、全実行時間に対して「ＰＥがアイドルである」である割合を「ＰＥアイドル率」と定義する．

「レディキューのアクセス競合が起こっている」とは、「レディキューに複数のＰＥがアクセスしようとしていて、ロック待ちが起こっている」状態を指す．同様に、「レディキューの競合率」を定義する．

2)負荷分散の方針

「ＰＥアイドル率」と「レディキューの競合率」を共に低く抑え、かつ、ゴールの分配が低コストで行えるように、3)に示すような負荷分散方式を行うことにした．これは、ＰＥの負荷の大きさをレディキューの長さで決めるという、単純な動的負荷分散である．（なお、newgoalとwakedgoalの区別も行っていない．）

3)負荷分散方式（以下ＬＢ２）

ＰＥは各自のレディキューを持ち（このキューは他のＰＥからアクセス不可、その代わり、このキューのアクセスにロックは不要）、その他に全ＰＥからアクセス可能なエキストラキューというレディキューを１本設ける（このキューのアクセスには当然ロックが必要）．そして以下のようにして、自分のレディキューまたはエキストラキューにゴールをエンキューする．

```
if (ipn != 0) then
    エキストラキューにエンキュー
else if (ready_len > extra_len/n)
    エキストラキューにエンキュー
else
    自分のレディキューにエンキュー
```

ipn:　アイドルプロセッサの数
ready_len:　自分のレディキューの長さ
extra_len/n:　エキストラキューの長さのｎ分の１
（データは全てｎ＝４）

4)負荷分散方式の評価

3)で述べた負荷分散方式（ＬＢ２）を評価するために、次の単純な負荷分散方式（以下ＬＢ１）と比較した．

負荷分散方式（ＬＢ１）

ゴールキューは全体で１本で、各ＰＥがゴールを取り合う方式．もっとも単純な方式で、ＰＥのアイドル率は低く抑えられるが、レディキューへのアクセスが競合し、これが全体の性能低下につながる可能性がある．

5)測定結果

・ＰＥアイドル率（図３）
どちらの方式がよいか何とも言えない．また、両者とも、この値はわずかであった．

・レディキューのアクセス競合率（図４）
ＬＢ２のほうがＬＢ１よりもアクセス競合が少ない．

・実行時間（図５）
しかし、実行時間に大きな差はなかった．（「ＰＥアイドル率」や「レディキューのアクセス競合率」は、性能に関してそれほど大きな要因ではなかったと言える．）

6．おわりに

密結合マルチプロセッサ上にＫＬ１の並列処理系を試作し、台数効果、処理系の各部分のＣＰＵの占有率などを測定した．また、本処理系の負荷分散方式ではレディキューの競合を低く抑えられること、ＰＥのアイドル率やレディキューの競合は性能にそれほど大きな影響を与えないなどの結果が得られた．

謝辞

日頃御指導いただいている棚橋部門長、林部長並びに、ＰＩＭ研究開発メンバに感謝いたします．

参考文献

［１］清水他：密結合マルチプロセッサへの並列処理系の実装、第３４回情処全国大会２Ｑ－１

［２］佐藤他：密結合マルチプロセッサでの並列処理系の評価、第３４回情処全国大会２Ｑ－２

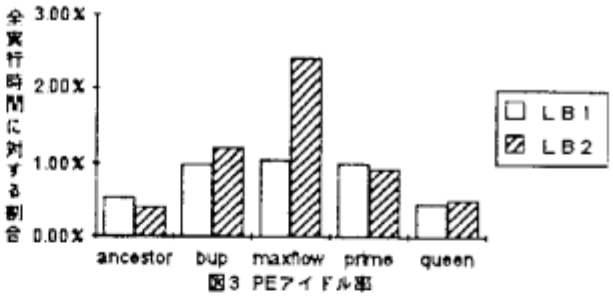

図3 PEアイドル率

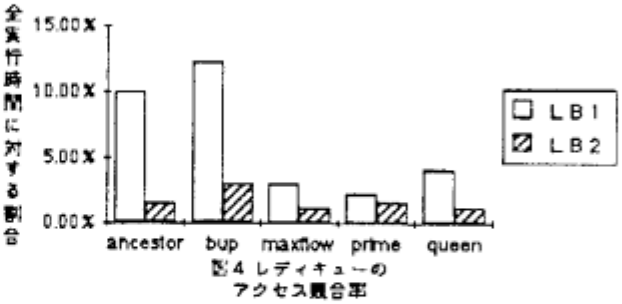

図4 レディキューのアクセス競合率

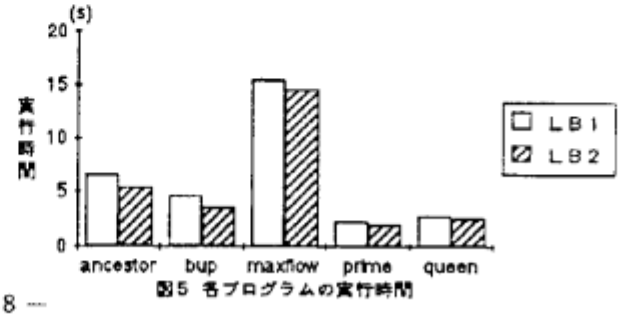

図5 各プログラムの実行時間