2H−6 クラスタ間メモリ共有型アーキテクチャにおけるKL1処理系の考察

酒井 浩, 仲瀬 明彦, 武脇 敏晃†
(株) 東芝 総合研究所　†情報通信システム技術研究所

1 はじめに

並列推論マシンPIMでは、要素プロセッサやネットワーク・トポロジを異にする数種類のアーキテクチャが提案され、研究開発の途上にある[1]。これらのマシンでは、およそ、メモリを共有する複数の要素プロセッサでひとつのクラスタを構成し、それら複数のクラスタがネットワークで結合されている。

PIMの任意のふたつのクラスタ(A,Bとする)の間では、負荷分散のためのゴールの受渡しや値の参照/書き込みを行う必要があるが，その実現手段としては、下記の2通りの方法が考えられる。

- クラスタAのプロセッサが、クラスタBのプロセッサにメッセージを送り、クラスタB内での処理を依頼する。(以下では、**クラスタ間分散メモリ方式**と呼ぶ)
- クラスタAのプロセッサが、クラスタBの共有メモリに直接アクセスする。(以下では、**クラスタ間共有メモリ方式**と呼ぶ)

PIMでは、これまでクラスタ間分散メモリ方式を前提に、並列論理型言語KL1用のハードウェア及び処理方式等が論じられてきた。本稿では、クラスタ間共有メモリ方式におけるKL1用ハードウェア及び処理方式について考察を行う。

なお、本研究は通産省第五世代コンピュータプロジェクトの一環として行った。

2 ハードウェアに関する考察

クラスタ間共有メモリ方式で各プロセッサからマシン内のすべての共有メモリに高速にアクセスするには、共有メモリにユニークなアドレスづけを行う必要がある。また、各プロセッサのロード/ストア命令で共有メモリをアクセスするには、ウイークリオーダが保証されている必要がある[1]。クラスタ間共有メモリ方式のハードウェアとしては、下記の2通りの方式が考えられる。

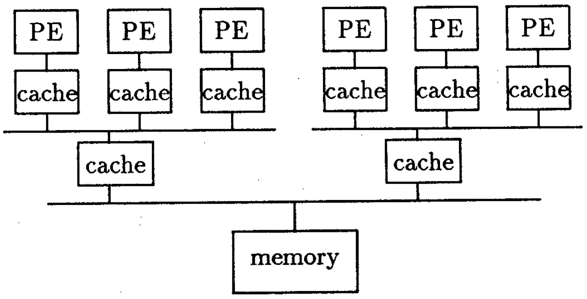

図 1: クラスタ間共有メモリ方式 (1)

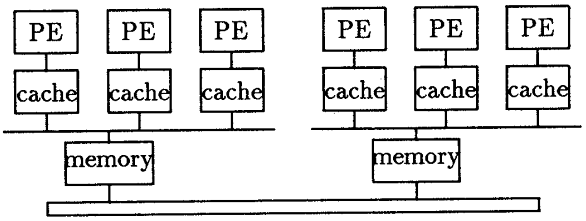

図 2: クラスタ間共有メモリ方式 (2)

- 図1に示すように並列キャッシュを階層化する。この方式では、各クラスタへのメモリの割付けを動的に変えられるのが利点である。反面、共有メモリ側のバスがネックとなるのを避ける必要があり、そのためには共有メモリに近い側のキャッシュのサイズをクラスタのヒープサイズと同じ程度まで大きくする必要がある。また、バストランザクションを減らすため、ダイレクトライトのようなプロトコルがサポートされている方が望ましい。
- 図2に示すように、要素プロセッサからそのクラスタの共有メモリへのアクセスには通常の並列キャッシュを使用し、他のクラスタの共有メモリへのアクセスには共有バスまたはスイッチングネットワークを使用する。後者へのアクセスでは、データをキャッシングする必要は特に無いと考える。その理由は、ゴールレコードの受渡しでも一般のオブジェクトの参照でも、一度アクセスしたものを繰り返し参照する可能性は低いためである。このアーキテクチャでは、各クラスタへのメモリの割り付けを動的に変えるのは(クラスタ間でのメモリアクセス回数が増え

---

[1] 分散メモリ方式のPIMでは、ウイークリオーダも保証されない(あるクラスタから到着したメッセージの順序が必ずしも送出した順であるとは限らない)ネットワークを扱うため、クラスタ間にまたがるデータの単一化の処理などが複雑化している。

KL1 System on the Intercluster Shared Memory
Hiroshi SAKAI, Akihiko NAKASE, Toshiaki TAKEWAKI†
Toshiba Corporation

るため)難しい。反面、キャッシュを階層化する必要がなく、ハードウェアが複雑になるのを避けることができる。

また、クラスタ間での負荷分散や一括 GC を小さなオーバーヘッドで実現するには、一般の PIM で採用されているようなスリットチェック機構が望ましい。

3 処理方式に関する考察

3.1 クラスタ間でのデータ共有

クラスタ間分散メモリ方式の場合、他のクラスタ中のオブジェクトを間接的に指し示すため、輸出入表が提案されている。これは、例えば、あるクラスタ A 内のポインタが別のクラスタ B 内のデータを指す場合に、クラスタ A のポインタはクラスタ A 内の輸入表の該当エントリを指し、そのエントリにはクラスタ B 内の輸出表の対応するエントリを指し、そのエントリには、本来のポインタが指すべきアドレスが格納される。

クラスタ間共有メモリ方式の場合、他のクラスタ中のオブジェクトを指し示すのに輸出入表を使用する方式とクラスタ内のデータと同様に直接アドレスで扱う方式の2通りが可能である。前者の後者に対する効果及び費用は下記の通りである。

- 効果:
  1. 各オブジェクトについて、クラスタ外からの参照度数を管理することにより、参照されなくなったデータの領域を速やかに再利用できる。
  2. ひとつのクラスタ単独での一括 GC が可能である。
- 費用:
  1. 輸出入表の管理のオーバーヘッドは1回のメモリアクセスと比較して、ひと桁程度大きいと思われる。また、その処理プログラムは、タイミングに関する部分が多く、デバッグが難しい。
  2. 他クラスタのデータを参照するためのデータタグが増加する。

著者らは、上記の考察の結果として、クラスタ間共有メモリ方式の場合には、輸出入表を使わず、直接アドレスで指し示せば良いと考えている。下記では、これを前提として考察を進める。

3.2 クラスタ間での負荷分散

クラスタ間分散メモリ方式の PIM では、暇なクラスタが他のクラスタに対してゴールを要求することにより、負荷分散を行う。

クラスタ間共有メモリ方式は、次のような特徴を持つ。

- 長所:
  1. クラスタ間で自動負荷分散を行う場合には、負荷の軽いクラスタから他のクラスタの負荷を直接調べることが可能である。
  2. あるクラスタは、別のクラスタのゴールをその要素プロセッサを煩わせることなしに奪うことが可能である。また、ゴールの引数としてポインタが格納されている場合、3.1 で述べた理由により、処理が軽くなる。
- 短所:
  1. クラスタ間転送の負荷が、ゴールをメッセージとしてバースト転送する場合より、かなり大きくなる。

3.3 一括 GC に対する考察

一般に PIM では、クラスタ内のヒープを消費した場合、一括 GC を行う必要がある。

クラスタ間共有メモリ方式では、あるクラスタ単独での GC は不可能である。それは、輸出入表が無いため、どのメモリセルがごみであるか、クラスタ内の情報のみでは判定できないためである。従って、全クラスタでの一括 GC を行う必要がある。一括 GC の方式は、クラスタ間分散メモリ方式の場合に考えられているような生きているゴールをマーキングルートとするコピーイング GC [2] で良い。ただし、他クラスタのオブジェクトを指しているところでは、輸出入表のような構造を一時的に作るなどして、クラスタ間のアクセスを抑える必要がある。

クラスタ間共有メモリ方式では、一括 GC と同時にオブジェクトをクラスタ間で移動することにより、クラスタ渡りのポインタの数を減らしたり、各クラスタでのメモリ消費量を均等化することができると考えられる。

従って、一括 GC のタイミングとしては、下記の2通りが有効と考えられる。

1. ヒープ領域が消費された時。
2. クラスタを結合するバスの混雑度合いが、あるしきい値を越えた時。

4 おわりに

クラスタ間でメモリを共有するような並列推論マシンを想定し、ハードウェア及び処理方式を考察した。今後は、一括 GC の方法などについて更に検討を進めるとともに、並列推論マシン PIM/k への適用を予定している。

参考文献

[1] A.Goto et al, "Overview of the Parallel Inference Machine Architecture (PIM)" in Proceedings of the FGCS88, 1988.

[2] 佐藤 正俊、後藤 厚宏,「KL1 処理系の評価 − メモリ消費特性と GC −」, JSPP89 予稿集, 1989.