KLIC 分散メモリ処理系におけるメッセージ通信の実現と評価

仲瀬 明彦、六沢 一昭、近山 隆
(財) 新世代コンピュータ技術開発機構

1 はじめに

KLIC[1] は、移植性と実行効率を重視した並列論理型言語 KL1 処理系であり、KL1 プログラムを C プログラムに変換し、コンパイル・実行することにより高い移植性と実行効率を実現している。

KLIC 分散メモリ処理系は、KLIC 逐次処理系核に分散処理用ライブラリを付加することにより実現する。このことにより、逐次実行時の効率を維持したまま移植性のある KLIC の分散実装が可能となる。

本稿では、KLIC 分散メモリ処理系におけるメッセージ通信の実現方式と試験実装の評価について述べる。

2 KLIC 分散メモリ処理系の実装方式

KLIC 分散メモリ処理系では、図 1に示すように、KLIC の逐次処理系核に分散処理管理部とメッセージ通信処理部を付加したプロセスが複数実行される。これらのプロセス間では、各プロセスの分散処理管理部とメッセージ通信処理部を経由して通信が行なわれる。

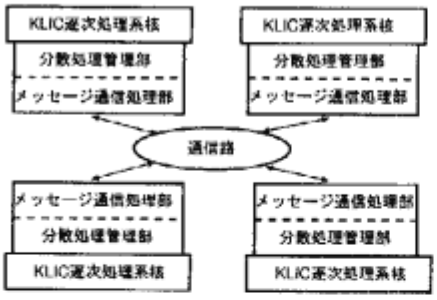

図 1: KLIC 分散実装の基本方式

KLIC 分散メモリ処理系の、分散処理管理部分は、並列推論マシン PIM で用いた分散処理方式を KLIC 用に再構成した方式を使用する [2]。

分散処理管理部分は、KLIC のジェネリックオブジェクト [1] と呼ばれる機能拡張機構を使って実現される。この方式では、KLIC の逐次処理系核を変更せずに分散メモリ処理系を構築できるので、逐次実行時の効率を低下させずに分散実行することができる。

メッセージ通信処理部分では、各種の通信ライブラリや通信方式に対して移植性を高めるため、基本通信操作を表 1に示すものに限定した。

表 1: KLIC 分散処理系の基本通信操作

送信処理	受信処理
送信バッファの作成	受信バッファの作成
送信バッファに書き込み	受信バッファに受信
送信バッファの送信	受信バッファから読み込み

メッセージ通信処理部から呼び出されるこれらの処理はマクロで定義しておき、使用する通信ライブラリや通信方式によってマクロの定義を書換えることにより、移植性の高い分散実装を実現する。

3 分散メモリ処理系の試験的実装

今回は試験的実装として、汎用の通信ライブラリを使った実装と共有メモリを通信路とした実装を試みた。

3.1 汎用の通信ライブラリを使った実装

汎用通信ライブラリを使った試験的実装では、PVM[3] を用いてメッセージ通信を実現した。

PVM では、KLIC 分散処理系の基本通信操作が、C/Fortran から呼び出せるライブラリ関数として提供されており、これらを用いることにより容易にメッセージ通信が実現できる。

また、PVM は種々のアーキテクチャに対応しており、異なるアーキテクチャのマシンを結合することも可能なので、KLIC の移植性を高める上で有利である。

3.2 共有メモリを通信路とした実装

共有メモリ型並列マシン上で KLIC 分散メモリ処理系を実行する場合は、共有メモリを高速通信路として使用すれば、通信オーバヘッドの削減が期待できる。

共有メモリを用いた試験的実装では、各プロセスの通信用バッファを共有メモリ上に用意し、その通信バッ

Message Passing Scheme of Distributed KLIC System
Akihiko Nakase, Kazuaki Rokusawa, Takashi Chikayama
ICOT, Mita Kokusai Bldg.21F, 4-28 Mita 1-chome, Minato-ku, Tokyo 108 Japan

ファを排他的にアクセスするように基本通信操作マクロを書換えることにより、メッセージ通信を実現した。

4 初期評価結果

4.1 単一 PE での実行時の性能

PVM を通信ライブラリとして使った KLIC 分散メモリ処理系を 1 PE で実行させたときの性能と、KLIC 逐次処理系の性能の比較を行なった。測定は SparcStation10/50 (CPU: SuperSPARC 50MHz) 上で行なった。測定結果を表2に示す。プログラム名に続く括弧内の数字は、ベンチマークプログラムに与えられたパラメータである。測定に用いたベンチマークプログラムの詳細は [4] を参照のこと。

表 2: 単一 PE での実行時間 (単位 秒)

プログラム	逐次版 KLIC	分散版 KLIC
nrev (100000)	16.8	16.7
queens (12)	18.2	18.2
primes (50000)	19.0	19.0
qsort (100000)	20.9	20.9

逐次処理系核を変更せずに分散処理を実現しているため、KLIC 分散メモリ処理系を 1PE で実行させた時の速度は、KLIC 逐次処理系の速度とほぼ同じである。

4.2 並列効果

次に、並列実行を行なった時の並列効果について述べる。評価プログラムとして、13-queens, 14-queens プログラムを用いた。

この queens プログラムでは、盤面の 2 列目まで駒を置いた状態の副問題群を各プロセッサに順に割り付けて行き、最後に 1 つのプロセッサが全プロセッサからの解の数を集計して解を出力する。メッセージ通信は、主に最初の問題分割部分と最後の解収集部分で発生するが、問題のサイズが大きくなるほど計算時間当たりのメッセージ数は少なくなる。

表3に、PVM を通信ライブラリとして使った KLIC 分散メモリ処理系の実行時間 (計算を開始してから全プロセッサが計算を終了するまでの時間) と並列効果を示す。測定は、SparcStation10/50 (CPU:SuperSPARC 50MHz) 8 台をネットワークで接続したシステム上で行なった。

8 台程度までの分散であれば、まずまずの並列効果が出ている。14-queens に比べて 13-queens の並列効果が若干悪くなるのは、PVM を用いた通信のオーバーヘッドが大きいため、問題の大きさ当たりの通信量の大きさがボトルネックになっているからである。

表 3: PVM を通信路とした時の実行時間 (単位 秒)

プロセッサ数	1	2	4	8
13-queens	118.4	62.9 (1.9)	35.4 (3.3)	18.9 (6.3)
14-queens	811.3	437.4 (1.9)	222.9 (3.6)	114.7 (7.1)

表 4: 共有メモリを通信路とした時の実行時間 (単位 秒)

プロセッサ数	1	2	4	8
13-queens	159.2	80.9 (2.0)	45.6 (3.5)	21.8 (7.3)
14-queens	1114.4	566.9 (2.0)	291.3 (3.8)	154.4 (7.2)

表4に、共有メモリ型並列マシン上で、共有メモリを通信路として実装した場合の並列効果を示す。測定は、SparcCenter2000 (CPU: SuperSPARC 40MHz) で行なった。

13-queens, 14-queens ともに、同じような並列効果が出ている。これは、共有メモリ経由の高速な通信を行なっているため、ある程度までは、通信量の増大によるオーバヘッドが押えられているからである。

5 おわりに

KLIC 分散メモリ処理系の実現方式と、その試験的実装の評価を述べた。

KLIC 分散メモリ処理系を単一 PE で実行した時の性能は、KLIC 逐次処理系の性能とほぼ同じで、問題の種類により高い並列効果も得られている。

しかし高速の通信路が確保できないと、問題によってはかなりのオーバヘッドが生じる危険性があるので注意が必要である。

今後は、より大規模な並列マシンへの移植や、プログラムの静的解析による通信量／通信頻度の削減等が望まれる。

参考文献

[1] Chikayama et al.: *A Portable and Efficient Implementation of KL1*, PLILP'94, 1994.

[2] 六沢, 仲瀬, 近山: KLIC 分散処理系の開発, 第 49 回情報処理学会全国大会 1T-8, 1994.

[3] Geist, Beguelin, Dongarra, Jiang, Manchek, Sunderam: PVM 3 USER'S GUIDE AND REFERENCE MANUAL, ORNL/TM-12187, May 1994

[4] 近山, 藤瀬, 関田, 中村: KLIC 処理系核の評価, 第 49 回情報処理学会全国大会 1T-6, 1994.