分散型大規模行列計算におけるデータ管理法

1L−4

松村 博光　大鎌 広　藤原 祥隆
北見工業大学

1　はじめに

ネットワークに接続された複数の計算機で分散処理を行なうためのソフトウェアとして、PVM[1]やMPI[2]が知られている。

本研究で提案している分散処理システム[3]は、用途を行列計算に限定することでより簡単で拡張性の高いユーザインタフェイスを実現している。

分散処理システムの形態はマルチサーバ・1クライアント型であり、クライアント・サーバ間、及びサーバ・サーバ間において行列データの通信を行なっている。

このような分散処理を行なう場合、計算機間のデータ通信にかかる時間がシステム全体の性能に大きな影響を与えることになる。このため、計算機間の通信を極力少なくするような行列データの管理法が要求される。

そこで、分散処理システムに適した行列データの管理法の設計を行なう。

2　システム概要

分散処理に用いる計算機は、その役割によって行列計算クライアントと行列計算サーバに分けられる。

＜行列計算クライアント＞

ユーザはまず本システムで提供するC++による分散行列クラスライブラリを用いて行列計算のプログラムを作成する(これをユーザプログラムとする)。以下にユーザプログラムの簡単な例を示す。

```
#include "DMatrix.h"
void main(void)
{
    DMatrix a("data1.dat"),
            b("data2.dat"), c;
    c = a + b;
    c.Print("kekka.dat");
}
```

Data management of distributed computation for large-scale matrix
Hiromitsu MATSUMURA, Hiroshi OHKAMA, Yoshitaka FUJIWARA
Dept. of Computer Sciences Kitami Institute of Technology
Koen-cho 165 Kitami-shi 090

行列計算クライアントでは、このユーザプログラムが実行される。ユーザプログラムはライブラリによって実行時に行列計算サーバへの命令列に変換され、各サーバに送信される。命令には大きく分けて行列データ通信命令、計算命令、特殊命令がある。例えば上記のプログラムはおおよそ次のような命令に変換される(行列計算サーバが4台の場合)。

- 行列aを小行列1～4に分割し、各サーバに送信
- 行列bを小行列5～8に分割し、各サーバに送信
- 行列cを小行列9～12に分割し、各サーバに送信
- サーバ1に加算命令(小行列1+小行列5→小行列9)を送信
- サーバ2～4にも同様に加算命令を送信
- 各サーバに計算結果(小行列9～12)を送り返すよう命令

＜行列計算サーバ＞

クライアントが送信してくる命令を実行するため、予め計算サーバプログラムが動作している。サーバプログラムの処理は基本的に以下のようになっている。

1. 命令受信
2. 命令デコード
3. 命令が終了命令なら処理を終了
4. 命令実行
5. 1.に戻る

4.の命令実行は行列計算だけでなく、データ送信も含まれる。また、処理の高速化のためにデータフロー型アルゴリズムを用いたり送信処理の子プロセス化[3]を行なっている。

このように、1台の行列計算クライアントが各行列計算サーバに行列データや計算命令を送信し、それを受信した行列計算サーバが命令を実行することで、分散処理が成り立っている。

3　行列データの管理

分散処理システムでの行列データの管理は、ユーザプログラム内で使用される行列データを行列計算サー

バ数で分割した行列 (以後分割された行列のことを小行列と呼ぶ) を単位に行なわれる。

行列計算クライアントでは、ユーザプログラム内で分散行列クラスのオブジェクトが生成されると直ちにその行列データを分割して小行列を作成する。この際、各小行列に対して全プログラム実行時間においてシステム全体でユニークな ID (行列 ID) が与えられる。

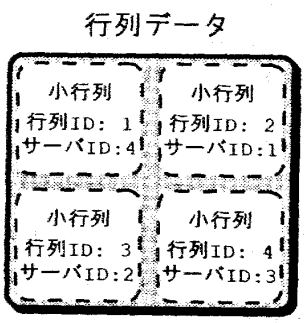

図 1: 小行列による行列データ管理

この行列 ID を用いることで、行列計算クライアント、サーバ双方で効率的な行列データの管理が可能となっている。

3.1 行列計算クライアントでの行列データ管理

クライアント側では、小行列を作成するとすぐに各行列計算サーバに送信する。送信後はユーザプログラム内の行列データと小行列の対応関係を記録して、行列データは解放してしまう。つまり行列データは基本的にサーバ側で保持され、クライアントは必要になったときのみサーバから受信する。こうすることでクライアント・サーバ間の通信を減らすことができ、結果としてクライアントの負荷を軽減できる。

クライアントでは計算命令も送信するが、計算命令の基本的なフォーマットは以下のようになっている。

- 命令の種類 (加算、減算、乗算など)
- 被演算子 1 の行列 ID
- 被演算子 2 の行列 ID

このとき、サーバ側に必要な行列データが全て揃っていないことも考えられる。もしクライアント側で、被演算子となる小行列が計算命令の送信先のサーバに格納されているかを予め確認できれば、確認の結果、足りない小行列を格納しているサーバに対して行列データの送信命令を発行するだけでよいことになる。

このため小行列に対しては、行列 ID だけでなくどの行列計算サーバに格納されているかを示す行列計算サーバ ID も記録しておく。この行列 ID と行列計算サーバ ID によって、クライアントでは小行列がどこの行列計算サーバに格納されているかを把握できる。

3.2 行列計算サーバでの行列データ管理

行列計算サーバで扱う行列データは小行列だけである。つまり、サーバ側では小行列とユーザプログラム内の行列データとの対応を、全く意識しない。ただしクライアントとは違い、行列 ID だけでなく実際のデータも格納する。

ここで、クライアントからの指示はすべて行列 ID を用いて行なわれるため、小行列のデータを行列 ID で検索できるような形で格納しなければならない。

そこで小行列を格納するためのリスト (行列リスト) を作成した。行列リストでは、行列 ID をキーとして小行列データの挿入、削除、検索が可能になっている。

このように、行列計算サーバ側での行列データの管理はクライアントと比較して非常に単純にしている。このような設計を行なったのは、サーバの主な処理が行列計算と行列データ通信であり、行列データの管理を複雑化するとこれらの処理の高速化の弊害になるためである。

4 まとめ

最後に、SGI 社の O2(R5000 180MHz) を行列計算サーバに 4 台、行列計算クライアントに 1 台使用し、1500 × 1500 の行列同士の乗算を行なったときの計算時間を示す。

分散処理しない	258 秒	サーバ 4 台	69 秒

このように、サーバ 4 台で分散処理を行わない場合の約 3.7 倍という結果が出た。

今後はサポート演算の追加、疎行列の対応などを中心に研究を進めていきたい。

参考文献

[1] Al Geist, etc:"PVM 3 USER'S GUIDE AND REFERENCE MANUAL", 1993.

[2] Message Passing Interface Forum:"MPI: A Message-Passing Inrerface Standard", 1995.

[3] 松村 博光, 大鎌 広, 藤原 祥隆:"分散型大規模行列計算における通信ブロックの軽減", 電気関係学会北海道支部連合大会講演論文集, p.353(1997-10)