

並列計算機環境とデータ解析について

小宮由里子、南 弘征、水田正弘
北海道大学 情報メディア教育研究総合センター

1 はじめに

データ解析分野で扱うデータ量は年々増大しまたそれらデータを扱う手法は複雑化しており、そのコンピュータ処理において、計算の効率化は重要な課題の一つであります。特に反復計算を要するものについては計算を並列化することによって全体の計算時間を短縮することが期待出来ます。しかし並列計算のために、並列計算機能を持つスーパーコンピュータを利用することは、一般のユーザにとって価格、プログラミング技術の点で容易なことではありません。

近年、ネットワーク環境が充実し、多くのコンピュータがネットワーク接続されるようになっていきます。またコンピュータマシン性能の向上には目覚ましいものがあります。それらネットワーク接続された個々のマシンをネットワークを通して結合し、仮想的な一つのシステムとみなし、そこで大きな計算を並列的に実行させるという提案がなされています。そのようなシステム構築のための提案として、MPI (Message Passing Interface)、PVM (Parallel Virtual Machine) などがあります。

本報告ではPVMを利用した並列計算機環境について紹介し、クラスタ分析に対する計算を適用例として述べます。

2 PVM環境

PVMは米国 Oak Ridge 国立研究所によって開発され提供されているフリーソフトウェアであり、利用者は Web から自由にソフトウェアをダウンロードし、自分のマシンにインストールすることが出来る。

PVMの特徴を次に挙げる。

- TCP/IP 接続された複数のマシンを仮想的な一つのシステムとみなす。
- PVM 環境を構成する各マシンには PVM ライブラリと PVM デーモンがインストールされていなければならない。
- マスター / スレーブモデル。
- システム内に OS が異なる (UNIX, Linux, Microsoft Windows NT/2000) マシンを混在させることが出来る。
- 現在アプリケーションプログラミング言語として、C, C++, Fortran がサポートされている。

3 PVMによるクラスタ分析

一般に非階層的クラスタ分析において、データの初期条件（初期クラスタリング）は、結果として得られるクラスタリングに対し大きく影響する。従って最適なクラスタリングを得るために、初期条件を変更しながら、クラスタリングを得る計算を繰り返す必要がある。ここに PVM マスターが複数のスレーブへ初期条件を配送しながら、スレーブのクラスタリング計算を制御するような並列計算アルゴリズムを考えることが出来る。

3.1 k - means 法アルゴリズム

Code 1: クラスタ分析のマスター用 C 言語擬似コード

```
1:  /* Master host */
2:
3:  #define NPROC "タスク数"
4:  #define CONDITION "初期条件数"
5:
6:  data をデータファイルから読み込む;
7:
8:  /* ↓ここから時間測定開始 */
9:
10: mycondition = CONDITION / NPROC; /* 1 タスク当りの初期条件数 */
11:
12: for(i=0; i<NPROC; i++){
13:     Slave  $S_i$  に data を送信する;
14:     Slave  $S_i$  に 乱数の元  $seed_i$  を送信する;
15:     Slave  $S_i$  に 1 タスク当りの初期条件数 mycondition を送信する;
16: }
17:
18: /* クラスタリング結果が送られてくるのを待つ */
19:
20: for(i=0; i<NPROC; i++){
21:     Slave  $S_i$  から  $Sw_{min_i}$  を受信する;
22:     Slave  $S_i$  から クラスタリング結果  $c\_result_i$  を受信する;
23: }
24:
25: NPROC 個の  $Sw_{min_i}$  の中から、最小のものを選び出し
26: それに対応するクラスタリング結果  $c\_result_i$  を
27: クラスタリング結果として採用する;
28:
29: /* ↑ここで時間測定終了 */
30:
31: クラスタリング結果を出力する;
```

Code 2: クラスタ分析のスレーブ用 C 言語擬似コード

```
1:  /* Slave  $S_i$  */
2:
3:  master から data と 乱数の元  $seed_i$  を受信する;
4:  for(j=0; j<mycondition; j++){
5:      初期条件を設定する;
6:      data を  $k$  - means 法を用いてクラスタリングする;
7:  }
8:
9:  規定回数クラスタリングした中から クラスタ内平方和の和  $S_w$  を
10:  最小にするクラスタリング結果  $c\_result_i$  と
11:  その最小  $S_w$  である  $S_{w\_min_i}$  を選択する;
12:
13:  master に  $S_{w\_min_i}$  を送信する;
14:  master に  $c\_result_i$  を送信する;
```

3.2 数値実験

3,000 点からなる 2 次元データ (図 1) に対してクラスタ数 5 としてクラスタリングを行った。初期条件は 1,000 個、従ってクラスタ計算を 1,000 回繰り返した。

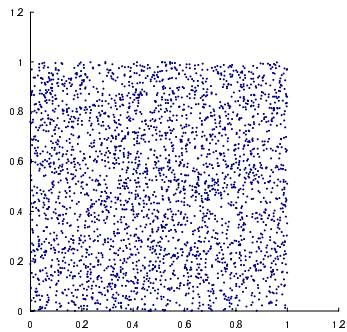


図 1 : データ点

実験に用いたコンピュータは次の通りですべて同一仕様とした。PVMはバージョン 3.4 を用いた。

CPU ... Pentium III (450MHz)

OS ... Linux RedHat 6.0

メモリサイズ ... 192MB

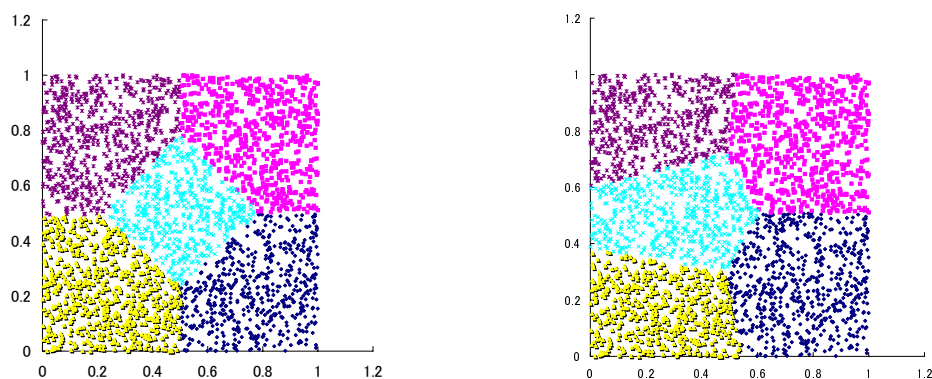


図 2: 図 1 に示したデータを 5 つのグループにクラスタリングした結果. 最小 S_W が得られたときのクラスタリング結果 (左図) と最大の S_W のときのクラスタリング結果 (右図)

表 1. に実験を 10 回行い測定した実行時間の平均値と分散、およびマシン数増加による速度向上比を示す。

表 1. 実行時間 (10 回実行の平均/分散) と速度向上比

ノード数	1	2	3	4	5	10	25	50
時間 (秒)	260.5	143.2	87.8	65.2	60.8	30.2	13.8	7.4
分散	0.3	0.2	0.8	0.8	0.2	0.2	0.2	0.2
速度向上比	1	1.82	2.97	4.00	4.28	8.63	18.88	35.20

ノード数 (マシン数) 50 の実験で、1 台による場合の約 35 倍の速度向上が得られた。

4 おわりに

PVM による並列計算を身近なマシンを利用して行った。データ解析分野で他に膨大な計算が必要となる Bootstrap 法、射影追跡層別逆回帰法などに適用した報告がなされている。これらを含めデータ解析分野でさらに有効な利用が期待出来る。

参考文献

Geist, A., Beguelin, A., Dongarra, J., Jiang, W., Manchek, R. and Sunderam, V.(1994). *PVM: Parallel Virtual Machine, A User's Guide and Tutorial for Networked Parallel Computing*, MIT Press.

Kawane, M., Komiya, Y. and Mizuta, M.(2000). Cluster Analysis with Parallel Virtual Machine. *Proceedings of The Tenth Japan and Korea Joint Conference of Statistics*, Beppu, Japan, pp.101–106.

http://www.epm.ornl.gov/pvm/pvm_home.html
(米国 Oak Ridge 国立研究所, Web サイト).