

# F-1-6

## PVMによる並列処理の高速化

00-1-26-058 宮内 翔  
(情報論理工学研究室)

### 1. 序論

複雑、膨大な量の計算には並列計算が用いられる。しかし並列計算機は高価であり計算機自体の性能拡張も容易ではないため、決して利用しやすいものではない。

そこで、ネットワークで接続された計算機の集合を仮想の並列計算機として用いるクラスタ処理が費用、機能拡張性といった点で優れており、最小のコストで大規模な問題を解くことができるため、現在並列計算の主流となっている。クラスタは、複数の計算機とそれを結ぶネットワークから成り、各計算機はネットワークを通じて互いに通信しながら並列に処理を行う。クラスタ処理のためのソフトとしてPVM (Parallel Virtual Machine)<sup>[1]</sup>、MPI (Message-Passing Interface)<sup>[2]</sup>等がある。

PVMはクラスタ処理を行うための実装パッケージであり、計算機の集合を単一の大規模並列計算機として用いることができる。また、MPIは計算機間のインタフェースを規定しており、この規定に従うことにより計算機間の協調が可能となる。

PVMは異機種種の計算機間の通信を前提としているため、PVMを実装する計算機の機種やOSを統一する必要は無い。この点はPVMを使う利点の一つである。MPIでは異機種間での通信は仮定されていないため、場合によっては機種、OSを統一する必要がある。

本研究では、PVMをWindows上に実装するに当たり、その方法と実装中および使用中に起こり得る問題点について調査および考察を行う。

### 2. 研究内容

PVMの構築を行うためには、まずPVMのソースファイルが必要となる。PVMはフリーソフトであり、現在、<http://www.netlib.org/pvm3/>等、様々なサイトからUNIX、あるいはWindows用のPVMのソースファイルをダウンロード可能である。PVMはデーモンとライブラリから成っており、PVMを構築する全ての計算機上でデーモンを作る必要があるため、それを構築する全ての計算機にソフトをダウンロードする必要がある。ダウンロードし、コンパイルした後、PVMを実行させるために環境設定、主に環境変数の設定とパスの指定が必要となる。このとき、計算機の機種、OSが異なれば環境設定も異なるため、個々の計算機ごとにその環境に応じた値に設定する必要がある。

PVM上でのプログラミングは、現在、C、C++、FORTRAN、JAVAといった言語で可能である。これらの言語中で、計算機間でのメッセージ通信を行う命令を記述することにより並列計算が可能となる。

### 3. 結果・考察

PVMはそれを構築する各計算機がデーモンを作りTCP/IP通信を行う。このため比較的大きな帯域幅のネットワークが必要とされる。

ネットワークの帯域幅が小さい場合、通信に大きな時間が掛かり、結果として計算機1台よりもかえって多くの時間が掛かる可能性もある。従って、PVMの利用には十分な帯域幅を持つネットワークを準備せねばならない。

並列処理を行うためには並列アルゴリズムが必要である。PVMは分散メモリ型の並列計算機であるため、PRAM (Parallel Random Access Machine)<sup>[3]</sup>の並列アルゴリズムをPVM上で実行させた場合、効率良く実行とは限らない。従ってPVMを使用する場合は、BSP (Bulk-Synchronous Parallel Machine)<sup>[4]</sup>モデルやCGM (Coarse Grained Multi-Computer)<sup>[5]</sup>モデルといったPVMに適応したモデル上で並列アルゴリズムを設計する必要がある。

### 4. 結論

本研究では、PVMをWindows上に実装するに当たり、その方法と実装中および使用中に起こる得る問題点について調査および考察を行った。今後の課題としては、起こり得る種々のエラーに対し、それらの原因と対処法を機種、OS別に分かり易く文章化することが考えられる。

### 参考文献

- [1] 村田英明: “PVM3.4 & リファレンスマニュアル,” (1995)
- [2] P. S. Pacheco, 秋田博(訳): “MPI並列プログラミング,” 培風館(2001)
- [3] Joseph JáJá: “An Introduction to Parallel Algorithms,” Addison-Wesley (1992)
- [4] L. G. Valiant: “A Bridging Model for Parallel Computation,” Communication of the ACM, Vol. 33, No. 8, pp. 103- 111 (1990)
- [5] F. Dehne, A. Fabri and A. Rau-Chapman: “Scalable Parallel Computational Geometry for Coarse Grained Multicomputers,” Proceeding of ACM Symposium on Computational Geometry, pp. 298- 307 (1993)