

PCグリッド上でのMATLAB分散計算と評価

小西 克巳[†], 中沢 亨[‡]

[†]工学院大学CPDセンター

[‡]大日本印刷株式会社

概要

本稿では数値計算ソフトウェアであるMATLABの分散計算をPCグリッド上で実現するミドルウェアを紹介する。このミドルウェアを利用することで、既存のネットワークとデスクトップPCを用いて最大255台のPCによるMaster-Worker型の並列計算を実現することが可能である。またMATLAB Compilerにより作成された実行ファイルを配布することで、MATLABのライセンス数が1つしか必要ないという特徴を持つ。本稿ではMATLABの分散計算の評価実験として、連続ウェーブレット変換、制御系設計問題における分枝限定法および遺伝的アルゴリズムを扱い、ミドルウェアの有効性を示す。

Performance Evaluation of Parallel Computing Tool for MATLAB on PC Grid Environment

Katsumi KONISHI[†] and Tohru NAKAWAZA[‡]

[†]Kogakuin University

[‡]Dai Nippon Printing Co., Ltd.

ABSTRACT

This work presents grid and parallel computing middleware, which enables MATLAB users to develop a parallel program on PC grid computing environment. The middleware uses no PC cluster and no special device, but needs some personal computers and only a MATLAB license for a single computer (master computer), even though we can use up to 255 worker computers. Most serial MATLAB programs, which are developed for control design, signal processing, image processing, financial modeling and analysis, etc., can be parallelized in a few steps if they can be compiled by the MATLAB Compiler, which convert MATLAB programs into stand-alone applications. This paper shows some examples of the continuous wavelet transform, a branch and bound method and a genetic algorithm for PID controller design.

1. はじめに

数値計算ソフトウェアであるMATLABは、そのプログラミングの容易性と汎用性から多くのユーザが利用している。特に自動車産業や計測制御産業、信号処理の分野ではデファクトスタンダードな開発環境であり、MATLABで開発された多くの研究開発資産がある。また、工学の分野のみではなく、医療や生物、金融に関する研究

分野でも利用が進んでいる。このような状況で、MATLAB上で並列計算を実現する要望は非常に高い。

MATLAB環境で並列計算を実現するミドルウェアやTOOLBOXは、すでに数種類が開発され公開されている²⁾³⁾。PCクラスタ上でPVM/MPIを実装したものや、UNIXのファイルシステムを利用して擬似的にMPIを利用可能としているものなどがある。特に後者はMATLAB

以外の専用ソフトウェアを必要としないため、ネットワークや並列計算の専門知識のない技術者にとって利用し易いものとなっている。また MATLAB の開発発売元の MathWorks 社から Distributed Computing Toolbox が発売されているが、日本での販売は未定である。

これらのすでに公開されているミドルウェアや TOOLBOX は、並列計算を行う PC の全てに MATLAB がインストールされている必要がある。つまり、PC の台数と等しい数の MATLAB ライセンスを所有している必要があり、大規模な研究機関以外での利用は難しい。また MATLAB 環境とは別に並列計算環境をユーザが構築する必要があり、並列計算環境に関する専門知識を持たない技術者が利用する障害となっている。

著者らは大日本印刷株式会社が開発し販売している PC グリッドミドルウェア “AD-POWERs¹⁾” を基に MATLAB で並列分散計算を実現するミドルウェアを開発した。AD-POWERs は MS Windows が搭載された PC 上で動作するグリッドミドルウェアで、科学技術計算分野では可視化システムなどにも用いられている⁴⁾⁵⁾⁶⁾。専用のネットワークやデバイスを必要とせず、各 PC にインストールするだけで容易に PC グリッド計算環境を構築できるという特徴を持つ。

本稿では MATLAB 対応版 AD-POWERs について紹介し、性能評価を行う。性能評価実験では、MATLAB が主に利用される分野である信号処理と制御系設計分野に注目し、連続ウェーブレット変換とコントローラ設計問題における分枝限定法と遺伝的アルゴリズムの計算実験を行う。

2. PC グリッド計算環境

本研究では、大日本印刷株式会社が開発し販売している PC グリッドミドルウェア AD-POWERs¹⁾を基に MATLAB 上で並列分散計算を実現するミドルウェアを開発している。AD-POWERs は Master-Worker 方式で動作するミドルウェアで、MS Windows 上で動作し、サーバとなる PC が全てのデータ・通信を管理し、実際に計算を行う worker PC に計算指示を出す。また、マルチキャスト通信を基本とした通信を用いることで、クライアントソフトウェアが起動している PC を自動的にキャッチすることが可能となし、サーバへのクライアント PC の登録や管理といった手間が不要という利点がある。クライアント PC 用のソフトウェアはスクリーンセーブで起動可能であるため、Windows 標準のスクリーンセーブ機能で遊休となっている PC の判断

が可能であり、スクリーンセーブ ON で計算処理に参加、OFF で離脱することが可能である。このため、計算処理に参加するクライアント PC の台数が動的に変化するという特徴を持つ。

AD-POWERs による並列分散計算の動作は、およそ以下の通りである。

Master PC

1. 同一 LAN 内の worker PC にクライアントソフトウェアの起動状況を問い合わせる。
2. クライアントソフトウェアが起動している worker PC に対し、全ての PC に同一の dll ファイル、および、個別のデータファイル（フォルダも可）を送信する。
3. worker PC からの計算結果を待ち、1 台の worker PC から結果が得られる毎に結果処理を行う。
4. 条件を満たしていれば終了し、満たしていなければ 1 - 3 を繰り返す。

Worker PC

1. master PC から dll ファイルとデータファイルを受け取る。
2. dll ファイルを実行し、結果および結果データファイルを master PC に返す。

上記の動作のうち master PC の 3 の動作では、1 台の worker PC から結果が得られる毎に処理が可能のため、非同期的に動作する並列プログラミングが可能である。また、全ての worker PC が同期して動作することも可能である。

3. MATLAB 上での並列分散計算の実現

本節では、前節で消化した AD-POWERs を利用した MATLAB 上での分散並列計算の実現について述べる。

MATLAB では C 言語等で開発した外部アプリケーションを MATLAB Compiler を利用して mex ファイルと呼ばれる MATLAB 実行ファイルにすることで、MATLAB から関数として呼び出すことができる。この機能を利用し、AD-POWERs の機能を MATLAB 関数として実装した。関数の具体的な説明は第 4 節で述べる。また、MATLAB Compiler は MATLAB で作成した関数を実行ファイル (exe ファイル) にすることが可能で、MATLAB 対応版 AD-POWERs でも、この機能を利用する。

MATLAB 上での分散計算は次のように実現される。まず、各 worker PC で計算させたい関数を MATLAB 上で作成し、MATLAB Compiler を利用して実行ファイル (exe ファイル) を作成

する。次に、MATLAB から mex ファイル関数として AD-POWERs を呼び出す。AD-POWERs の機能により、master PC から worker PC へ dll ファイル、MATLAB で作成した実行ファイル、データファイルが転送される。ここで、dll ファイルには、MATLAB で作成した実行ファイルを実行するという内容の命令が書かれている。これにより、各 worker PC 上では MATLAB で作成した実行ファイルが実行され、結果として MATLAB で開発した関数が全ての worker PC で実行されることにより並列計算が実現される。AD-POWERs と同様に、結果は master PC に集められ、実行結果は MATLAB で読み込み可能な mat ファイルとして MATLAB に渡される。このときの動作概念図を Fig.1 に示す。

MATLAB Compiler で作成した実行ファイルは自由に配布可能であるため、MATLAB ライセンスを必要としない。これにより、master PC 用の MATLAB ライセンスのみで 255 台までの並列計算が可能となる。

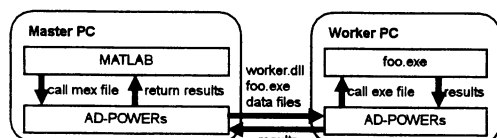


Fig.1 MATLAB 上での並列分散計算の概念図

4. MATLAB でのコマンド例

本節では、MATLAB 対応版 AD-POWERs で実装されている MATLAB 関数のうち主要なものを紹介する。ここで紹介する関数は、全て master PC 用の関数である。worker PC では MATLAB Compiler で作成された実行ファイルが実行されるだけであるので、worker PC 用の関数は存在しない。Worker PC では、プログラムは通信等を意識せず、結果を返す関数として MATLAB 言語で作成すれば良い。また、具体的な利用法を説明するため、本節では連続ウェーブレット変換での具体例を示す。

master PC から各 worker PC に配布するファイルを指定するには **ADP_initMaster** を利用する。

```
[ rtn0 , rtns0 ] = ADP_initMaster('SendFiles');
```

SendFiles はファイル名であるが、フォルダを指定することで、フォルダごと配布することができる。

各 worker PC に計算タスクを割り振るには **ADP_assignTask** 関数を用いる。

```
[ rtn , rtns ] = ADP_assignTask(TIME_OUT,
                                JOB_ID, JOB_NAME, EXE_FILE,
                                FILE_1, FILE_2, ... , FILE_N);
```

上記の EXE_FILE は MATLAB Compiler で作成した実行ファイル名であり、FILE_N は各 worker PC に配布するデータファイル名である。このとき、このデータファイル名の個数により、並列分散計算を行う worker PC の台数が決定する。worker PC の台数よりも計算タスクの数の多い場合は、計算が終了した worker PC に再度計算タスクを割り当て、全ての計算タスクが終了するまで繰り返す。

現在の計算タスクの実行状況を確認するには、**ADP_Check** を用いる。

```
[rtn , rtns] = ADP_Check();
```

返り値が 0 のとき、全ての計算は終了したことを表し、1 のときは計算中を表す。予期しないエラーで計算が中断されている場合は 9 を返す。この関数により実行状況が把握できる。

全ての worker PC での計算を強制的に終了させたい場合は、**ADP_Stop** を用いる。

```
[rtn , rtns] = ADP_Stop();
```

全ての worker PC からの結果を得たい場合には、**ADP_getResult** を用いる。

```
[rtn , rtns] = ADP_getResult();
```

この関数は全ての計算タスクの結果が得られるまで待ち続ける。よって、予期しないエラーで worker PC での計算が停止している場合には、**ADP_getResult()** 関数は値を返さずに停止した状態になる。これを避けるには、**ADP_Check** 関数を利用すれば良い。

次に連続ウェーブレット変換での具体例を示す。連続ウェーブレット変換は、MATLAB の Wavelet Toolbox を利用することで以下のように実現できる。

```
c = cwt(s,scale,'db3');
```

上式により、信号 s は 3 次の Daubechies フィルタによって連続ウェーブレット変換される。 $scale$ は離散化されたスケールを表す。本来、連続ウェーブレット変換では全てのスケールに対

し値を持つが、計算機上では実現不可能であるため、スケールを離散化しベクトル **scale** として与えて計算する。具体的には以下のように用いる。

```
scale = [ 1 : 0.005 : 128 ];
c = cwt(s,scale,'db3');
```

このとき、**scale** を分割することで並列計算による連続ウェーブレット変換の高速化が行える。つまり、master PC ではスケールを分割しておき、worker PC では与えられたスケールと信号に対して **cwt** 関数を利用して連続ウェーブレット変換し、master PC で結果を統合すれば良い。

worker PC 用の MATLAB プログラムは以下のようになる。

```
function [result] = cwt_worker();
```

% 省略 — 初期化など

```
load('signal.mat'); % load signal 's'
load(DataFileName); % load 'scale'
c_worker = cwt(s,scale,'db3');
```

% 省略 — ファイルクローズなど

簡単のために、上記プログラム中で初期化やファイルクローズ等の箇所は省略した。上記 m ファイルを MATLAB Compiler でコンパイルし、**cwt_worker.exe** を作成する。

master PC 用プログラムは以下のようになる。

```
% m-file for a master PC
```

% 省略 — 初期化など

```
s = waveread('sample.wav'); % set signal s
save('%SendFiles%signal.mat','s');
% save s as '%SendFiles%signal.mat'
[ rtn0 , rtns0 ] = ADP_initMaster('SendFiles');
```

```
scale_1 = [1:0.005:32]; save scale_1
scale_2 = [33:0.005:64]; save scale_2
scale_3 = [65:0.005:96]; save scale_3
scale_4 = [97:0.005:128]; save scale_4
```

```
[ rtn , rtns ] = ADP_assignTask(100,1,'wavelet',
'cwt_worker.exe', 'scale_1.mat','scale_2.mat',
'scale_3.mat','scale_4.mat');
```

```
[rtn2 , rtns2] = ADP_getResult();
```

% 省略 — ファイルクローズなど

上記プログラムでは、音声ファイル(sample.wav)を読み込み MATLAB 変数として格納し、**signal.mat** として保存して worker PC に配布している。1-128 の範囲のスケールを 4 分割し、それぞれを mat ファイルで格納し、**ADP_assignTask** 関数で worker PC への計算タスクとして割り振っている。その後、**ADP_getResult** 関数で計算結果を待つ。

このように、MATLAB 対応版 AD-POWERs を利用することで、MATLAB 上で容易に並列分散計算プログラムを記述できる。

5. 計算実験

5.1 実験環境

工学院大学 C P D センターラボラトリーの演習用 P C を利用して PC グリッド計算環境を構築した (Fig. 2)。サーバおよびクライアントとなる PC は全て同一機種であり、性能を Table 2. に示す。各スイッチには 18 台ずつの PC が 1 Gbps で接続され、スイッチ間は 4 Gbps でブリッジ接続されている。なお、master PC には MATLAB がインストールされているが、他の PC にはインストールされていない。

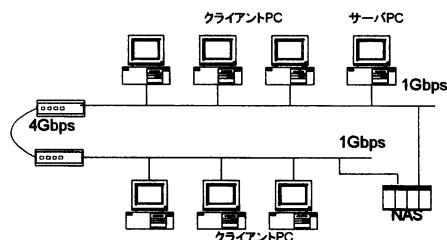


Fig. 2 実験環境

Table 3. 各 PC の性能

OS	: WindowsXP Pro SP2
CPU	: Pentium4 2.7GHz
Memory	: PC2700 512MB
HDD	: 40GB 5400rpm ATA100
NIC	: 1000Base-T

5.2 実験

MATLAB の代表的なアプリケーションとして信号処理および制御系設計が挙げられる。これらの具体例として、連続ウェーブレット変換の並列計

算, および, 制御系における PID コントローラの設計問題の並列解法の実験を行った.

連続ウェーブレット変換は前節で述べたように容易に並列計算に対応できる. 前節で示したプログラムを基に計算実験を行った. 3 秒分の音声ファイル (約 512kbyte) を 1-128 のスケールで連続ウェーブレット変換した. 結果を Table 4 に示す. なお, 1 台で実行したときの結果は, AD⁺POERS を利用せずに通常の MATLAB で行ったものである.

Table 4. 連続ウェーブレット変換の実験結果

1台で実行	7 3 5 sec
4 台	4 2 5 sec
8 台	2 6 0 sec
1 6 台	1 7 4 sec
3 2 台	1 2 1 sec

4 台で計算した結果を見ると, 1 台の PC で計算した場合の 2 倍の速さに達していない. これは連続ウェーブレット変換した結果は複素数であるため, 計算結果が変換前の信号の 2 倍の容量となることと, MATLAB 上で使用する mat ファイルは実際のデータ容量よりも大きくなるため, master PC と worker PC 間でのファイルの転送に時間を要したためである. 連続ウェーブレット変換の実験では, 全ての worker PC に同一のデータファイルを配布するため, worker PC の数が増えるとデータファイルの転送時間の影響が顕著に現れる. このため, 3 2 台の PC を利用しても 6 倍程度しか計算速度が向上しなかった. 実際に連続ウェーブレット変換を並列計算する場合は, スケール方向ではなく時間方向に計算を分割して行うため, 今回の実験よりも計算速度の向上が期待できる.

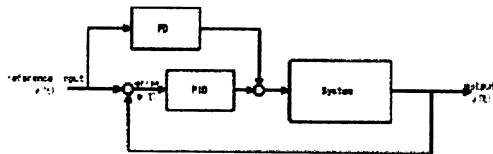


Fig. 5 PID 制御系

次に Fig. 5 に示すような PID コントローラ設計問題の実験を行った. PID コントローラは現在最も利用されている制御器である. この制御器の設計問題は双線形行列不等式 (bilinear matrix inequality, BMI) 問題と呼ばれる問題に定式化されるが, BMI 問題は NP 困難な問題⁷⁾で解くのは容易ではない. BMI 問題の解法として分枝限定法に基づく手法が提案されている⁸⁾⁹⁾. 分枝限定法は並列化が容易に可能で, PC クラスタ上

やグリッド上で並列分枝限定法を実装した研究¹⁰⁾も行われている. 本稿で扱う設計問題の詳細は文献 11) 12) を参照されたい.

MATLAB 上で並列分枝限定法を実装し, 実験を行った. ϵ 大域解と呼ばれる最適解が得られるまでに要した時間を Table 6 に示す.

Table 6. 分枝限定法の計算結果

1台で実行	611 min
4 台	192 min
8 台	92 min

連続ウェーブレット変換の際と同様に, 1 台で実行した結果は通常の MATLAB 上で行った結果である. 分枝限定法の場合は, master PC と worker PC との間でやりとりされるデータサイズが小さいため, 通信コストがほとんど発生しない. そのため, 8 台の PC で並列計算をした場合に 7 倍程度の計算速度を得ている. 分枝限定法は基本的には全領域探索アルゴリズムであるため, 並列計算に参加する PC の台数が増加するほど最適解を発見する確率が増加する. この点は通常の並列化計算の実験結果と異なることに注意されたい.

次に同様の PID コントローラ設計問題に対し, 遺伝的アルゴリズムで球解を行った. 遺伝的アルゴリズムも並列化が容易に実現できる. 本稿では, PID パラメータを求めるために単純 GA に基づく手法を用いた. BMI 問題に基づく手法では, 最適化計算により制御性能 (H2 ノルム) の計算をしているが, 本稿での遺伝的アルゴリズムでは, 与えられたパラメータを用いて実際に数値シミュレーションを行い, シミュレーション結果から制御性能を評価している. つまり, 1 個体につき 1 回のシミュレーションを行う. このとき, 個体数が増えるとシミュレーションに要する計算時間が増えるため, 遺伝的アルゴリズムにおける 1 世代の個体数を分割し, 各 worker PC でシミュレーションを行った.

遺伝的アルゴリズムは厳密解が得られる保証もなく, 得られた解の精度も保証されない. そのため, 世代数を固定して計算時間の比較を行った. 1 世代あたり 200 個体で 20 世代まで計算した計算結果を Table 7 に示す.

Table 7. 遺伝的アルゴリズムの計算結果

1台で実行	4061 sec
4 台	1296 sec
8 台	692 sec

遺伝的アルゴリズムの場合も分枝限定法の場合

と同様に, master PC と worker PC との間でやりとりされるデータサイズが小さいため, 通信コストがほとんど発生しない。このため, 並列化の効果が計算結果に表れている。

以上のように通常の他の並列分散計算を実現するミドルウェアと同様の結果が得られた。

5. おわりに

本研究ではPCグリッド計算環境でMATLABの並列分散計算を実現するミドルウェアを開発した。MATLABで並列計算を実現するミドルウェアやTOOLBOXは他にもあるが, 本研究で提供するミドルウェアは特別な環境を必要とせず, 1台分のMATLABライセンスしか必要としないという特徴を持つ。またMS Windows上で簡単に動作することから, 並列計算やネットワークの専門知識を持たない技術者も容易に利用することができる。

本稿では計算実験により, 開発したミドルウェアの性能を評価した。通常のPCクラスタやPCグリッドでの並列計算と同様の計算結果となり, 開発ミドルウェアの有効性が示された。

計算実験で示した分枝限定法および遺伝的アルゴリズムによる解法は, 1台のPCで動作する通常のMATLABプログラムとして開発されたものであり, 本研究で計算実験を行うために並列化したものである。このように, 本研究で提供するミドルウェアはプログラミングも容易で, 過去の開発資産を最大限に活用することが可能である。自動車産業や計測制御産業では, MATLABで開発されたプログラミング資産が多くあることから, 本研究で開発したミドルウェアにより, これらの分野での並列計算の実用化の発展が期待される。また, MATLABは制御や信号処理だけではなく, 金融工学や医療工学にも利用されているため, 様々な分野での並列計算の実用化に貢献できると考えられる。

参考文献

- 1) AD-POWERS: <http://www.ad-powers.jp/>
- 2) MPI Toolbox for MATLAB,
http://atc.ugr.es/javier-bin/mpitb_eng
- 3) Parallel Programming with MatlabMPI:
<http://www.ll.mit.edu/MatlabMPI/>
- 4) 中沢亨, 小西克巳, "PCグリッド「AD-POWERS」を用いた大量データ通信の最適化", 電子情報通信学会コンピュータシステム研究会, 東京, 2005年4月
- 5) 小西克巳, 中沢亨, 宮地英生, "PCグリッドによる大規模データのアニメーション可視化への応用", 可視化情報シンポジウム, 東京, 2005年7月(発表予定)
- 6) 宮地英生, 中沢亨, 小西克巳, "PCのGRID環境を使った大規模データの可視化", 計算工学講演会, 東京, 2005年5月
- 7) O. Toker and H. Ozbay, "On the NP-hardness of solving Bilinear Matrix Inequalities and simultaneous stabilization with static output feedback", Proc. of ACC, pp. 2525-2526, 1995
- 8) K. C. Goh and M. G. Safonov and G. P. Papavassilopoulos, "A Global Optimization Approach for the BMI Problem", Proc. of the 33rd Conf. of Decision and Control, pp. 2009-2014, 1994
- 9) H. Fujioka and K. Hoshijima, "Bounds for the BMI Eigenvalues Problem", Trans. of SICE, Vol. 33, No. 7, pp. 616-621, 1997
- 10) 合田 憲人, 二方 克昌, 原 辰次, "並列分散計算システム上でのBMI固有値問題解法", 情報処理学会論文誌ハイパフォーマンスコМПユーテイングシステム, 42(SIG12(HPS4)), pp.132-141, 2001
- 11) Jun Aoyama, Katsumi Konishi, Toru Yamamoto and Takao Hinamoto, "A BMI-Based Design for Robust PID Controllers with Two-Degrees-Of-Freedom Structure", Proc. of American Control Conference, pp. 4879-4884, Boston USA, July 2004
- 12) Jun Aoyama, Katsumi Konishi, Toru Yamamoto and Takao Hinamoto, "A BMI-BASED DESIGN OF SWITCHED PID CONTROLLERS", IFAC International Symposium on Advanced Control of Chemical Processes, Hong Kong China, January 2004