MPICH User’s Guide v4.0.2 翻訳 by DeepL

**1 Introduction**

本書は、MPICHがすでにインストールされていることを前提に書かれています。MPICH のインストール方法については、MPICH インストーラガイド、または MPICH のトップ・レベルのディレクトリにある README を参照してください。本書は、MPIアプリケーションのコンパイル・リンク・実行方法と、MPICHに付属 するツールの使用方法を説明します。このマニュアルは暫定版であり、まだ完成していない部分があります。しかし、MPICH を使い始めるには十分な内容になっていると思います。 MPICH を使い始めるには十分な内容です。

**2 Getting Started with MPICH**

MPICH は、MPI 規格の高性能で移植性の高い実装です。MPI-1, MPI-2, MPI-3 (動的プロセス管理、片側操作、並列入出力、その他の拡張機能を含む) のすべてを実装するように設計されています。MPICH インストーラガイド には、MPICH の設定とインストールに関するいくつかの情報があります。 MPIプログラムのコンパイル、リンク、実行に関する詳細は以下の通りです。

**2.1 デフォルトの実行環境**

MPICH はプロセス管理と通信の分離を実現します。MPICH のデフォルトの実行環境は Hydra と呼ばれます。他のプロセスマネージャーも利用可能です。

**2.2 並列処理の実行**

MPICHはmpiexecとその標準的な引数をすべて実装しており、さらにいくつかの拡張機能を備えています。mpiexec の標準的な引数についてはセクション 5.1 を、 様々なプロセス管理システムに特有の拡張についてはセクション5のサブセクションを参照してください。

**2.3 Fortranでのコマンドライン引数**

MPICH1(正確にはMPICH1のmpirun)では、Fortranを含むすべてのアプリケーションプログラムでコマンドライン引数にアクセスする必要とし、MPICH1の configureは、iargcとgetargの正しいバージョンを含むライブラリを見つけ、それらのライブラリをmpifort スクリプトが MPI プログラムをリンクするようにしています。MPICHはアプリケーションのコマンドライン引数へのアクセスを必要としないので、これらの関数はオプションであり、configureはこれらに対して特別なことはしません。もしアプリケーションでこれらが必要な場合は、使用する Fortran 環境でそれらが利用可能であることを確認する必要があります。使用する Fortran 環境で利用可能であることを確認する必要があります。

**3 Quick Start**

MPICHを使用するには、MPICHがインストールされているディレクトリを知る必要があります(あなた自身がインストールしたか、システム管理者がインストールしたかのどちらかです。この場合、/usr/localが一つの場所となります。MPICH がまだインストールされていない場合は、MPICH インストーラのガイド をご覧ください)。そのディレクトリの binサブディレクトリをパスに入れます。（訳注：ここでのパスはWindowsでいう環境変数Pathのこと）これにより、コンパイルやリンク、プログラムの実行のための様々な MPICH コマンドにアクセスできるようになります。binディレクトリにある他のコマンドは、ランタイム環境の一部を管理し、ツールを実行します。

最初に実行するコマンドのひとつは mpichversionで、MPICH の正確なバージョンと設定を知ることができます。このマニュアルに書かれていることのいくつかは、あなたが使っている MPICH のバージョンと、それがどのように設定されているかに依存します。

これで、MPIプログラムが実行できるようになったはずです。ここでは、MPICHをインストールしたディレクトリが /home/you/mpich-installed であるとします。そして、そのディレクトリをパスに追加したとします。そして、次のコマンドを実行します。

export PATH=/home/you/mpich-installed/bin:$PATH

そうしたら、1台のマシン上ではありますが、次のコマンドでMPIプログラムを実行することができます。（訳注：要するに、exampleディレクトリに公式によるサンプルプログラムが載っているということです）

cd /home/you/mpich-installed/examples

mpiexec -n 3 ./cpi

**4 Compiling and Linking**

これらのコマンドの詳細は後述しますが、ここで正常に実行できれば、MPICHが正しくインストールされ、MPIプログラムが実行できたことになります。

プログラムのコンパイルとリンクに便利な方法は、MPICH がビルドされたのと 同じコンパイラを使用したスクリプトを使用することです。そのコンパイラには、mpicc, mpicxx, mpifort があり、それぞれ C, C++, Fortran プログラム用です。もしこれらのコマンドのどれかが欠けている場合、それは MPICH がその言語をサポートせずに設定されたことを意味します。

**4.1 C++での問題点**

ユーザーによっては、次のようなエラーメッセージが表示されることがあります。

SEEK\_SET is #defined but must not be for the C++ binding of MPI

問題は、stdio.h （訳注：printfやscanfなどの標準的な関数を提供するCヘッダー）と MPI C++ インターフェイスの両方が SEEK SET, SEEK CUR, SEEK END を使っていることです。これは本当にMPI標準のバグです。そこでこれらをプログラムにでmpi.hのincludeの前に追加してみてください。

#undef SEEK\_SET

#undef SEEK\_END

#undef SEEK\_CUR

または以下の定義をコマンドラインに追加してください。

-DMPICH\_IGNORE\_CXX\_SEEK

これにより MPI バージョンの SEEK SET 等がスキップされます。

**4.2 Fortranでの問題点**

MPICHは、Fortranプログラムに対して2種類のサポートを提供しています。Fortran 77プログラマは mpif.h に MPI COMM WORLD などの MPI 定数の定義があります。Fortran 90プログラマは、代わりに MPI モジュールがあり、このモジュールは、多くの MPI 関数のインターフェイスを定義しています。しかし、この MPI モジュールは、Fortran 90 を完全にサポートしているわけではありません。たとえば、MPI Send のような "choice" 引数を取るルーチンのインターフェースは提供されていません。

**5 Running Programs with “mpiexec”**

MPI 標準規格では、MPI プログラムを実行するための推奨される方法として mpiexecが記述されています。MPICH は mpiexec 標準を実装しており、さらにいくつかの拡張機能を提供します。

**5.1 Standard “mpiexec”**

ここでは MPI Standard[1]にある標準的な mpiexecの引数を説明します。ローカルマシン上（単一マシン）で n個のプロセスを持つプログラムを実行するには、次のコマンドを使用します。

mpiexec -n <number> ./a.out　（訳注：a.outは実行ファイル）

複数ノード（ネットワークで接続された複数のマシン）で実行するには、次のコマンドを使用します。

Mpiexec -f machinefile -n <number> ./a.out

“machinefile”の書式は、次のようになります。

host1

host2:2

host3:4 # Random comments

host4:1

host1'、'host2'、'host3'、'host4'は、ジョブを実行するマシンのホスト名です。’2'、'4'、'1' のコロンの後の各数値は、各ノードで実行したいプロセスの数を表します。何も指定しない場合、':1' が指定されたものとみなします。

Hydraについてのさらに詳しい情報については、

<http://wiki.mpich.org/mpich/index.php/Using_the_Hydra_Process_Manager>

を参照してください。

**5.2 すべてのプロセス管理環境に対応する拡張機能**

いくつかの mpiexec の引数は、特定の通信サブシステム（「デバイス」）やプロセス管理環境（「プロセスマネージャ」）に特有のものです。私たちの意図は、すべての引数をデバイスとプロセスマネージャの間で可能な限り統一することです。以下、これらを別々に記述します。

**5.3 Hydra用のmpiexec拡張機能**

MPICHは、数多くのプロセスマネジメントシステムを提供しています。Hydra は MPICH のデフォルトのプロセスマネージャです。Hydra の詳細および mpiexec の拡張については、以下を参照してください。

<http://wiki.mpich.org/mpich/index.php/Using_the_Hydra_Process_Manager>

**5.4 gforker用の拡張機能**

gforker は一つのマシン上でプロセスを起動するためのプロセス管理システムで、 MPI プロセスが単に mpiexec プロセスから fork されることからこのように呼ばれています。このプロセスマネージャはMPI Commawn と他の動的プロセスルーチンを使用するプログラムはサポートしますが、 mpiexec で起動されないプログラムからの動的プロセスルーチンの使用はサポートしません。gforker プロセスマネージャは主にMPI プログラムの開発とテストを簡素化するため、デバッグの補助としてしようされます。

5.4.1 gforker用のmpiexecコマンドライン引数

mpiexec の標準的なコマンドライン引数に加え、gforkerのmpiexec は以下のオプションをサポートしています。

-np <num> 標準の -n 引数と同義です。

-env <name> <value> 　環境変数 <name> に <value> を設定します。

envnone 環境変数(他の -env または -genv 引数で指定されたもの以外)を渡さない。

環境変数(他の -env または -genv 引数で指定されたもの以外)を mpiexec によって実行されているプロセスに渡しません。

デフォルトでは、すべての環境変数が各 MPI プロセスに提供されます。

(根拠: ユーザが最も驚かない原則)

-envlist <list> リストされた環境変数(カンマで区切られた名前)を渡します。

カンマで区切られた名前) を現在の値と共に、mpiexec によって実行されるプロセスに渡します。

mpiexec で実行されているプロセスに現在の値を渡す。

-genv <名前> <値> オプションは同じ意味を持つ。

-genv オプションは対応する -env バージョンと同じ意味である。

ただし、現在の実行ファイルだけでなく、すべての実行ファイルに適用されます (コロン構文の場合)。

複数の実行ファイルを指定するためにコロン構文が使用されている場合)、現在の実行ファイルだけでなく、すべての実行ファイルに適用されます。

-genvnone -envnone と同様ですが、すべての実行ファイルに適用されます。

-genvlist <list> -envlist と同様ですが、すべての実行ファイルに適用されます。

-usize <n> 属性 MPI UNIVERSE SIZE の値として返される値を指定します。

-l 標準出力と標準エラー(stdout と stderr)に、プロセスのランクをラベル付けします。

プロセスのランク

-maxtime <n> タイムリミットを <n> 秒に設定する。

-exitinfo 以下の場合、各プロセスが終了した理由についての詳細情報を提供する。

異常終了がある

www.DeepL.com/Translator（無料版）で翻訳しました。