第3章 プロセス間通信の定番MPIを使って分散コンピューティング

ラズパイ・クラスタの実力を試す

宮田 賢一

表1　今回使ったOpen MPのC言語用API関数（Open MPIには400を超えるAPI関数がある）

関数名（C API）	mpi4pyでのメソッド名	意　味
MPI_Init	不要	MPI実行環境を初期化する
MPI_Comm_size	Get_size	コミュニケータに対する並列数を取得する
MPI_Comm_rank	Get_rank	コミュニケータ内での自プロセスのランクを取得する
MPI_Get_processor_name	Get_processor_name	自プロセスのプロセッサ名（通常ホスト名）を取得する
MPI_Finalize	不要	MPI実行環境を削除する
MPI_Bcast	Bcast	コミュニケータ内にデータをブロードキャストする
MPI_Reduce	Reduce	コミュニケータ内からデータを集約する
MPI_Wtime	Wtime	自プロセスが起動してからの経過時間を取得する
MPI_send	send	指定したプロセスにデータを送信する
MPI_recv	recv	指定したプロセスからデータを受信する

● 前章までで作った並列分散処理をCとPythonで試す

並列処理を使うために標準化された規格Open MPI（Message Passing Interface）はCとFortranのAPIを提供しています．Fortranは1950年代に登場した非常に歴史の古いプログラミング言語ですが，科学技術計算に向いた言語仕様を持つことや，数学関数のライブラリの蓄積が多数されていることから，スーパ・コンピュータを用いた計算分野であるハイ・パフォーマンス・コンピューティング（HPC）の世界では今でも現役のプログラミング言語です．

ただし本誌の読者にはFortranになじみのない方が多いと思いますので，Cによるサンプルを最初に示します．その後，Open MPIをPythonから使えるようにするmpi4pyパッケージを使ったMPIプログラムを紹介します．

Open MPIには400を超えるAPIが用意されています．**表1**に本稿で使用したC言語のAPI関数を抜粋しました．

CのAPI関数に対応するPythonのAPI（mpi4py）も比較のために追記しています．

APIの詳細仕様はウェブ・サイト(1)(2)を参照してください．それぞれの使い方はサンプル・プログラムと合わせて説明します．

これ以降，表現が曖昧にならない限り，通信規約であるMPIと，MPIの実装であるOpen MPIをともにMPIと表記することにします．

リスト1　MPIプログラミングの基本を把握するための最小構成プログラム（hello.c）

```
#include <mpi.h>
#include <stdio.h>

int main(int argc, char *argv[])
{
    int rank, size, name_len;
    char name[MPI_MAX_PROCESSOR_NAME];

    MPI_Init(&argc, &argv); /* 実行環境の初期化 */
    MPI_Comm_size(MPI_COMM_WORLD, &size);
                                /* 並列数を取得 */
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
                                                9
    MPI_Get_processor_name(name, &name_len);
                              /* プロセッサ名を取得 */
    printf("Hello, I'm process %d of %d on %s¥n",
                                rank, size, name);
    MPI_Finalize(); /* 実行環境の削除 */
    return 0;
}
```

最小プログラムで練習

MPIプログラミングの基本を把握しておくために，まずC言語を使って最小プログラムhello.cを作成します（**リスト1**）．このプログラムは次のコマンドに