

本マニュアルの目的

本マニュアルでは、単一ユーザが独占的に Linux サーバ (CentOS 6.6) を使用して OpenMX のジョブを並列実行するための環境構築方法と Winmostar のリモートジョブ投入機能から Linux サーバへジョブを投入する方法を示しています。計算環境は全てユーザのホームディレクトリ配下で行うことを想定しています。複数ユーザが使用する共用サーバの環境を構築する方法、複数ノードを利用する環境を構築する方法などは本マニュアルに含まれませんので、別途お問い合わせください。

なお、本マニュアルでは Linux サーバ上でジョブスケジューラ TORQUE が使用可能であると仮定しています¹。また以下の命名としています。

サーバホスト名 : remote_server

ユーザ名 : winmostar_user

I. インストール準備

- ① 以下のサイトにアクセスし、openmx3.8.tar.gz と patch3.8.4.tar.gz をダウンロードする。

<http://www.openmx-square.org/download.html>

Download of OpenMX

Available packages in terms of GNU-GPL

- **openmx3.8 (release date: 03/April/2016, 136 MB)** + **patch (05/Mar./2018)** README.txt
- openmx3.7 (release date: 23/May/2013, 112 MB) + patch (21/Feb./2015) README.txt
- openmx3.6 (release date: 10/Nov./2011, 079 MB) + patch (14/Nov./2011) README.txt

- ② 以下のサイトにアクセスし、fermisurfer_1.7.1.zip をダウンロードする。

<https://ja.osdn.net/projects/fermisurfer/releases/>



¹ <http://www.adaptivecomputing.com/products/open-source/torque/>を参照。管理者権限で yum などを用いて導入する。あるいは、X-Ability のサイト内 http://winmostar.com/jp/gmx4wm_jp.html (4.5. Torque) 参照。

- ③ openmx3.8.tar.gz, patch3.8.4.tar.gz と fermisurfer_1.7.1.zip を ftp など で Linux サーバ に 転送 する。こ
こではホームディレクトリ直下の
~/openmx3.8.tar.gz
~/patch3.8.4.tar.gz
~/fermisurfer_1.7.1.zip
に置いたと仮定する。
- ④ Linux サーバ に ログイン し、次 の コマンド を 実行 する こと で 必要 な パッケージ を インストール する。
- OpenMX のビルドに必要
- ```
$ yum install fftw3-devel lapack-devel blas-devel openmpi-devel gcc-g++ gcc-gfortran
```
- Fermisurfer のビルドに必要
- ```
$ yum install freeglut-devel.x86_64
```
- Winmostar から OpenMX を 実行 する 際 に 必要
- ```
$ yum install dos2unix
```
- ⑤ ~/.bashrc に 次 を 追加 する。
- ```
export PATH=/usr/lib64/openmpi/bin/:$PATH
export LIBRARY_PATH=/usr/lib64/openmpi/lib/:$LIBRARY_PATH
export PATH=${HOME}/openmx3.8/work/:$PATH
export OPENMX_DATA_PATH=~/.openmx3.8/DFT_DATA13
```
- ⑥ ~/.bashrc の 設定 を 有効 化 する。
- ```
$ source ~/.bashrc
```

## II. OpenMX3.8 のビルド

- ① 圧縮ファイルを解凍する。

```
$ cd
$ tar xzvf openmx3.8.tar.gz
```

- ② パッチをあてる。

```
$ cp patch3.8.4.tar.gz openmx3.8/source
$ cd openmx3.8/source
$ tar xzvf patch3.8.4.tar.gz
```

- ③ makefile を編集する。

```
$ vi makefile
```

139-141 行を以下のようにコメントアウトする。

```
#CC = mpicc -O3 -xHOST -ip -no-prec-div -openmp -I/opt/intel/mkl/include/fftw
#FC = mpif90 -O3 -xHOST -ip -no-prec-div -openmp
#LIB= -L/opt/intel/mkl/lib -mkl=parallel -lmkl_intel_lp64
-lmkl_intel_thread -lmkl_core -lpthread -lifcore -lmpi -lmpi_f90
-lmpi_f77
```

コメントアウトした直下に以下を書き加える。

```
CC = mpicc -fopenmp -O3 -I/usr/local/include
FC = mpif90 -fopenmp -O3 -I/usr/local/include
LIB = -L/usr/local/lib -lfftw3 -llapack -lblas -lmpi -lgfortran
-lmpi_mpihf -lmpi_usempi
```

- ④ コンパイルとインストールを行う。

```
$ make all
$ make install
```

- ⑤ 動作確認を行う。

```
$ cd ../work
```

```
$ openmx Cdia.dat
```

処理が終わって以下のメッセージが表示されたら正常に動作している。

```
The number of threads in each node for OpenMP parallelization is 1.
```

```

```

```

```

```
Welcome to OpenMX Ver. 3.8.3
```

```
Copyright (C), 2002-2014, T. Ozaki
```

```
OpenMX comes with ABSOLUTELY NO WARRANTY.
```

```
This is free software, and you are welcome to
```

```
redistribute it under the constitution of the GNU-GPL.
```

```

```

```

```

(途中省略)

|                 |   |   |       |   |       |
|-----------------|---|---|-------|---|-------|
| RestartFileDFT  | = | 0 | 0.000 | 0 | 0.000 |
| Mulliken_Charge | = | 0 | 0.001 | 0 | 0.001 |
| FFT(2D)_Density | = | 0 | 0.003 | 0 | 0.003 |
| Others          | = | 0 | 1.752 | 0 | 1.752 |

```
The calculation was normally finished.
```

### III. Fermisurfer のビルド

- ① 圧縮ファイルを解凍する。

```
$ cd
```

```
$ unzip fermisurfer_1.7.1.zip
```

- ② コンパイルとインストールを行う。

```
$ cd fermisurfer/src
```

```
$ make all
```

```
$ cp bxs2frmsf /usr/local/bin/
```

IV. Winmostar から GAMESS をリモートジョブ実行する。

- ① [Winmostar OpenMX 基礎編チュートリアル](#)の内容に従い操作を進めるが、キーワード設定ウインドウで **Run** ボタンを押さずに **OK** ボタンを押す。
- ② その後、[ユーザマニュアルのリモートジョブの実行手順](#)に従って操作を行い、**Get All Files** ボタンを押してファイルを取得するところまで実行する。
- ③ 再び [Winmostar OpenMX 基礎編チュートリアル](#)の内容に戻り、ローカルジョブの時と全く同じ操作方法で結果解析を行う。

以上