

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 (Rocky Linux 9.4) を使用して LAMMPS ジョブを並列実行するための環境を構築する方法と、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以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、GPU を用い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。

1. コンパイラの確認・インストール

- ① Linux サーバにログインし、以下のコマンドで make がインストールされているか確認する。

```
$ which make
```

インストールされていない場合は dnf などのパッケージ管理ソフトウェアを用いてインストールする。Rocky 9.4 で dnf を使う場合は以下のように実行する。

```
$ sudo dnf install make
```

- ② 以下のコマンドで gcc のバージョンが 4.4 以上であるか確認する。

```
$ gcc --version
```

gcc がインストールされていない、あるいは gcc のバージョンが 4.4 未満である場合は、dnf などのパッケージ管理ソフトウェアを用いてインストールする。Rocky 9.4 で dnf を使う場合は以下のように実行する。

```
$ sudo dnf install gcc gcc-c++ gcc-gfortran
```

2. MPI のインストール

dnf により MPICH をインストールする場合下記の手順で実行する。

- ① 下記のコマンドでインストールする

```
$ sudo dnf install mpich-devel
```

- ② 現在のシェルで Quantum ESPRESSO をビルドするために、以下のコマンドを実行して MPICH を有効にする。なお、~/.bashrc にこのコマンドを追記すると Linux サーバにログインし直すたびにこのコマンドの実行が不要となるが、同サーバ上における他のソフトウェアとの設定の調整が必要となる。本書では~/.bashrc への追記は必須ではない。

```
$ export PATH=/usr/lib64/mpich/bin:$PATH
```

```
$ export LD_LIBRARY_PATH=/usr/lib64/mpich/lib:$LD_LIBRARY_PATH
```

- ③ 以下のコマンドを実行してインストールした mpirun のパスが表示されることを確認する。

```
$ which mpirun
```

- ④ MPI の動作確認のため、以下のコマンドを実行して現在のディレクトリ名が 2 つ表示されることを確認する。

```
$ mpirun -n 2 pwd
```

他の MPI 環境と競合するなど問題が起こる場合は下記の手順に従いインストールする。

- ① ウェブブラウザで <https://www.mpich.org/static/downloads/3.1.4/mpich-3.1.4.tar.gz> をダ

- ダウンロードする。
- ② 保存した `mpich-3.1.4.tar.gz` を SCP, FTP など Linux サーバに転送する。ここではホームディレクトリ直下に置いたと仮定する。
 - ③ Linux サーバにログインし、圧縮ファイルを解凍する。

```
$ cd
$ tar xzvf mpich-3.1.4.tar.gz
```
 - ④ 以下の手順に従って MPICH 環境のセットアップを行う。詳しくは Installers' Guide (<http://www.mpich.org/static/downloads/3.1.4/mpich-3.1.4-installguide.pdf>) 参照のこと。

```
$ mkdir -p ~/mpich-install-3.1.4
$ mkdir -p ~/tmp/mpich-3.1.4
$ cd ~/tmp/mpich-3.1.4
$ ~/mpich-3.1.4/configure --prefix=${HOME}/mpich-install-3.1.4
$ make
$ make install
```
 - ⑤ 現在のシェルで LAMMPS をビルドするために、以下のコマンドを実行して MPICH を有効にする。なお、`~/.bashrc` にこのコマンドを追記すると Linux サーバにログインし直すたびにこのコマンドの実行が不要となるが、同サーバ上における他のソフトウェアとの設定の調整が必要となる。本書では `~/.bashrc` への追記は必須ではない。

```
$ export PATH=${HOME}/mpich-install-3.1.4/bin:$PATH
```
 - ⑥ 以下のコマンドを実行してインストールした `mpirun` のパスが表示されることを確認する。

```
$ which mpirun
```
 - ⑦ MPI の動作確認のため、以下のコマンドを実行して現在のディレクトリ名が 2 つ表示されることを確認する。

```
$ mpirun -n 2 pwd
```

3. LAMMPS のインストール

- ① LAMMPS のサイト内の[Downloads]サイトの old version のダウンロードページ
<https://download.lammps.org/tars/index.html> にウェブブラウザでアクセスし lammps-29Sep2021.tar.gz をダウンロードする。



```
2021-12-14 00:00 lammps-14Dec2021.tar.gz
2021-10-27 00:00 lammps-27Oct2021.tar.gz
2021-09-29 12:00 lammps-29Sep2021.tar.gz
2021-09-20 00:00 lammps-20Sep2021.tar.gz
2021-08-31 00:00 lammps-31Aug2021.tar.gz
```

- ② lammps-29Sep2021.tar.gz を SCP, FTP など Linux サーバに転送する。ここではホームディレクトリ直下に置いたと仮定する。
- ③ Linux サーバにログインし、以下のコマンドで圧縮ファイルを解凍する。なお、LAMMPS のバージョンによっては、拡張子が tar.gz になっても実際には gzip 圧縮されていないことがあるため注意する。

```
$ cd
$ tar xvf lammps-29Sep2021.tar.gz
```

- ④ 以下のコマンドで LAMMPS にパッケージを追加する。

```
$ cd lammps-29Sep2021/src
$ make yes-molecule yes-misc yes-rigid yes-reaxff yes-extra-dump yes-mc yes-kspace
yes-phonon yes-qeq yes-reaction yes-replica yes-dpd-basic yes-extra-molecule yes-
manybody yes-meam yes-extra-fix yes-fep yes-tally yes-colvars
```

- ⑤ 以下のコマンドでパッケージ設定を確認する。

```
$ make ps | grep YES
```

```
$ make ps | grep YES
Installed YES: package COLVARS
Installed YES: package DPD-BASIC
Installed YES: package EXTRA-DUMP
Installed YES: package EXTRA-FIX
Installed YES: package EXTRA-MOLECULE
Installed YES: package FEP
Installed YES: package KSPACE
Installed YES: package MANYBODY
Installed YES: package MC
Installed YES: package MEAM
Installed YES: package MISC
Installed YES: package MOLECULE
Installed YES: package PHONON
Installed YES: package QEQ
Installed YES: package REACTION
Installed YES: package REAXFF
Installed YES: package REPLICA
Installed YES: package RIGID
Installed YES: package TALLY
```

- ⑥ 以下のコマンドでシリアル実行用のモジュールをビルドする。

```
$ make lib-colvars args="-m serial"
$ make serial
```

- ⑦ 動作チェックのため以下のコマンドで LAMMPS をシリアル実行する。

```
$ ./lmp_serial < ../bench/in.lj >& log_serial.txt
```

実行後 log_serial.txt をテキストエディタで開き、末尾に以下のようなメッセージが表示されていれば、LAMMPS のバイナリが生成されていることになる。(環境・タイミングにより数値は若干異なる)

```
(途中省略)
...
Nlocal: 8000 ave 8037 max 7964 min
Histogram: 2 0 0 0 0 0 0 1 1
Nghost: 9007.5 ave 9050 max 8968 min
Histogram: 1 1 0 0 0 0 0 1 0 1
Neighs: 300708 ave 305113 max 297203 min
Histogram: 1 0 0 1 1 0 0 0 0 1

Total # of neighbors = 1202833
Ave neighs/atom = 37.5885
Neighbor list builds = 5
Dangerous builds not checked
Total wall time: 0:00:00
```

- ⑧ 以下のコマンドで並列実行用のモジュールをビルドする。

```
$ make lib-colvars args="-m mpi"
$ make mpi
```

- ⑨ 動作テストのため以下のコマンドで LAMMPS を並列実行する。

```
$ mpirun -n 2 ./lmp_mpi < ../bench/in.lj >& log_mpi.txt
```

実行後 log_mpi.txt をテキストエディタで開き、末尾に以下のようなメッセージが表示されていれば、LAMMPS のバイナリが生成されていることになる。(環境・タイミングにより数値は若干異なる)

```
(途中省略)
...
Step Temp E_pair E_mol TotEng Press
  0      1.44 -6.7733681      0 -4.6134356 -5.0197073
 100  0.7574531 -5.7585055      0 -4.6223613  0.20726105
Loop time of 0.880423 on 2 procs for 100 steps with 32000 atoms

Performance: 49067.319 tau/day, 113.582 timesteps/s
100.0% CPU use with 2 MPI tasks x no OpenMP threads

MPI task timing breakdown:
Section | min time | avg time | max time |%varavg| %total
-----|-----|-----|-----|-----|-----
Pair   | 0.72532  | 0.73676  | 0.75073  | 1.1 | 83.68
Neigh  | 0.083564 | 0.08495  | 0.085729 | 0.3 | 9.65
Comm   | 0.023556 | 0.03834  | 0.051227 | 5.3 | 4.35
Output | 6.6996e-05 | 7.2002e-05 | 7.7009e-05 | 0.0 | 0.01
Modify | 0.017242 | 0.017345 | 0.017438 | 0.1 | 1.97
Other  |          | 0.002952 |          |    | 0.34

Nlocal: 8000 ave 8037 max 7964 min
Histogram: 2 0 0 0 0 0 0 1 1
Nghost: 9007.5 ave 9050 max 8968 min
Histogram: 1 1 0 0 0 0 0 1 0 1
Neighs: 300708 ave 305113 max 297203 min
Histogram: 1 0 0 1 1 0 0 0 0 1

Total # of neighbors = 1202833
Ave neighs/atom = 37.5885
Neighbor list builds = 5
Dangerous builds not checked
Total wall time: 0:00:00
```

4. Linux サーバ上での最終確認

- ① Winmostar から接続するときの状況を再現するため、Linux サーバにログインしなおす。
- ② MPI の動作を確認するため、MPI の起動に必要な環境を設定するコマンドを入力しメモに控えておく。本書の手順に従った場合は、以下のコマンドを入力する。

```
$ export PATH=/usr/lib64/mpich/bin:$PATH
```

```
$ export LD_LIBRARY_PATH=/usr/lib64/mpich/lib:$LD_LIBRARY_PATH
```

または

```
$ export PATH=$PATH:${HOME}/mpich-install-3.1.4/bin
```
- ③ 以下のコマンドを実行してインストールした mpirun のパスが表示されることを確認する。確認できない場合は②の設定を見直す。

```
$ which mpirun
```
- ④ MPI の動作確認のため、以下のコマンドを実行して現在のディレクトリ名が 2 つ表示されることを確認する。確認できない場合は②の設定を見直す。

```
$ mpirun -n 2 pwd
```
- ⑤ LAMMPS の動作を確認するため、LAMMPS の起動に必要な環境を設定するコマンドを入力しメモに控えておく。本書の手順に従った場合は、以下のコマンドを入力する。

```
$ export PATH=$PATH:${HOME}/lammps-29Sep2021/src
```
- ⑥ 以下のコマンドを実行してインストールした LAMMPS のパスが表示されることを確認する。確認できない場合は⑤の設定を見直す。

```
$ which lmp_mpi
```
- ⑦ 以下のコマンドを実行して LAMMPS が起動しバージョンが表示されることを確認する。確認できない場合は⑤の設定を見直す。LAMMPS が起動したら Ctrl-c で終了する。

```
$ mpirun -n 2 lmp_mpi
```

表示される例：

```
LAMMPS (29 Sep 2021 - Update 3)
```

5. Winmostar からの設定および動作確認

ユーザマニュアルの [7.2. リモートジョブの設定手順](#)に従って設定し動作確認を行う。

テンプレートスクリプトを編集する際には、「# Insert commands here」 から 「# Do not modify the followings」 の間に 4-②と 4-⑤でメモに控えた内容を追記する。また、「# Insert commands here」 から 「# Do not modify the followings」 の間に OPT_LAMMPS_CONV_RESTART という変数の設定がデフォルトで書かれている。利用している LAMMPS のバージョンに合わせ必要な行を残し、行頭にコメントアウト (#) がある場合は削除する。本書の手順に従った場合は以下ようになる。

```
echo "*****"
echo "***      Set user-defined variables      ***"
echo "*****"
set -v
# Insert commands here
MPI_COMMAND="mpirun -n %WM_NUM_PROC% "
BIN_LAMMPS=lmp_mpi

OPT_LAMMPS_CONV_RESTART=-restart2data # for LAMMPS after 24Oct2018

export PATH=$PATH:${HOME}/mpich-install-3.1.4/bin
scl enable devtoolset-7 bash
export PATH=$PATH:${HOME}/lammps-29Sep2021/src

# Do not modify the followings
```

以上