

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 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。

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、サーバ名とユーザ名が以下であると仮定しています。

サーバ名 : remote_server

ユーザ名 : winmostar_user

I. Linux サーバ上の LAMMPS の動作環境設定方法

- 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

<http://lammps.sandia.gov/tars/> にウェブブラウザでアクセスし lammps-30Jul16.tar.gz をダウンロードする。²

	lammps-24Jul17.tar.gz	24-Jul-2017 09:02 94MGZIP compressed docume>
	lammps-24Mar17.tar.gz	24-Mar-2017 15:16112MGZIP compressed docume>
	lammps-26Jan17.tar.gz	26-Jan-2017 17:03100MGZIP compressed docume>
	lammps-28Jun14.tar.gz	07-Aug-2014 07:39 62MGZIP compressed docume>
	lammps-28Mar17.tar.gz	28-Mar-2017 13:26102MGZIP compressed docume>
	lammps-30Jul16.tar.gz	30-Jul-2016 12:42 90MGZIP compressed docume>
	lammps-30Mar18.tar.gz	30-Mar-2018 14:39131MGZIP compressed docume>
	lammps-30Oct14.tar.gz	11-Nov-2014 08:15 57MGZIP compressed docume>

- ② lammps-30Jul16.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の ~/lammps-30Jul16.tar.gz に置いたと仮定する。

- ③ Linux サーバにログインし環境を確認する。

make, 及び gcc (4.4 以降) が利用可能となっている必要がある。以下のコマンドで確認する。

```
$ make --version
```

```
$ gcc --version
```

環境がない場合、あるいはバージョンが古い場合は、yum などのパッケージ管理ソフトを用いて導入する。

- ④ 圧縮ファイルを解凍する。拡張子は tar.gz になっているが実際には gzip されていないこともあるため注意する。

```
$ cd
```

```
$ tar xvf lammps-30Jul16.tar.gz
```

¹ <http://www.adaptivecomputing.com/products/open-source/torque/> を参照のこと。管理者権限で yum 等を用いて導入する。あるいは、クロスアビリティのサイト内の http://winmostar.com/jp/gmx4wm_jp.html (4.5. Torque) を参照のこと。

- ⑤ make を用いてビルド環境を設定する³。

```
$ cd ~/lammps-30Jul16/src
$ make yes-misc
$ make yes-rigid
$ make yes-user-reaxc
```

- ⑥ ビルド環境を確認する。

```
$ make ps
```

```
$ make ps
  Installed NO: package ASPHERE
  Installed NO: package BODY
  Installed NO: package CLASS2
  Installed NO: package COLLOID
  Installed NO: package COMPRESS
  Installed NO: package CORESHELL
  Installed NO: package DIPOLE
  Installed NO: package GPU
  Installed NO: package GRANULAR
  Installed NO: package KIM
  Installed NO: package KOKKOS
  Installed YES: package KSPACE
  Installed YES: package MANYBODY
  Installed NO: package MC
  Installed NO: package MEAM
  Installed YES: package MISC
  Installed YES: package MOLECULE
  Installed NO: package MPIIO
  Installed NO: package OPT
  Installed NO: package PERI
  Installed NO: package POEMS
  Installed NO: package PYTHON
  Installed NO: package QEQ
  Installed NO: package REAX
  Installed NO: package REPLICA
  Installed YES: package RIGID
  Installed NO: package SHOCK
  Installed NO: package SNAP
  Installed NO: package SRD
  Installed NO: package VORONOI

...
  Installed NO: package USER-QUIP
  Installed YES: package USER-REAXC
  Installed NO: package USER-SMD
...
```

- ⑦ ビルドする

```
$ make serial
```

- ⑧ ~/.bashrc に次の一行を追加する。

```
export PATH=${HOME}/lammps-30Jul16/src:$PATH
```

II. LAMMPS 並列版の動作環境設定方法 (LAMMPS の並列実行を行わない場合は、必要無い) ⁴

- ① MPICH サイトにアクセスする。

<http://www.mpich.org/downloads/>

³ LAMMPS が提供する make は様々な機能を有する。\$ make ↵ で機能が表示される。

⁴ yum などのパッケージ管理ソフトにより MPICH をインストールしても構わない。

- ② mpich-3.1.4 をダウンロードする。
- ③ 保存した mpich-3.1.4.tar.gz を 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く。

- ④ 圧縮ファイルを解凍する。

```
$ cd
$ tar xzvf mpich-3.1.4.tar.gz
```

- ⑤ 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
$ mkdir -p ~/tmp/mpich-3.1.4
$ cd ~/tmp/mpich-3.1.4
$ ~/mpich-3.1.4/configure --prefix=${HOME}/mpich-install
$ make
$ make install
```

- ⑥ ~/.bashrc に次の一行を追加する。
export PATH=\${HOME}/mpich-install/bin:\$PATH

- ⑦ LAMMPS の並列実行モジュールをビルドする

```
$ cd ~/lammcs-30Jul16/src
$ make mpi
```

- ⑧ 動作チェックを行う⁵。

```
$ cp lmp_mpi ../bench
$ cd ../bench
$ mpirun -np 4 lmp_mpi < in.lj >& log.txt
```

log.txt をテキストエディターで開く。

⁵ ここでは、1 ノード内での 4 コア並列実行を想定している。

```

:
:
:
Lattice spacing in x,y,z = 1.6796 1.6796 1.6796
Created orthogonal box = (0 0 0) to (33.5919 33.5919 33.5919)
  1 by 2 by 2 MPI processor grid
Created 32000 atoms
Neighbor list info ...
  1 neighbor list requests
  update every 20 steps, delay 0 steps, check no
  max neighbors/atom: 2000, page size: 100000
  master list distance cutoff = 2.8
  ghost atom cutoff = 2.8
  binsize = 1.4, bins = 24 24 24
Setting up Verlet run ...
  Unit style : lj
  Current step: 0
  Time step : 0.005
Memory usage per processor = 4.09506 Mbytes
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 4 procs for 100 steps with 32000 atoms

Performance: 49067.319 tau/day, 113.582 timesteps/s
100.0% CPU use with 4 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 1 0 1
Neighs: 300708 ave 305113 max 297203 min
Histogram: 1 0 0 1 1 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

```

上記のとおりメッセージであれば正常にコンパイルとリンクが正しく行われている。

III. Winmostar から LAMMPS をリモートジョブ実行する方法

- ① [Winmostar LAMMPS 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 LAMMPS 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。

IV. (追補) MEAM などの package を追加する場合の動作環境設定方法

LAMMPS では MEAM などのポテンシャル関数は、前述の標準的な環境設定のみでは使用できない。以下に示す手順が必要となる。

- ① 追加 package に必要となるライブラリのコンパイルを行う⁶

```
$ cd ~/lammps-30Jul16/lib/meam
```

```
$ make -f Makefile.gfortran
```

- ② make を用いてビルド環境を設定する。

```
$ cd ../src
```

```
$ make yes-meam
```

- ③ LAMMPS のモジュールをビルドする

```
$ make serial [並列実行の場合]
```

```
$ make mpi [並列実行の場合]
```

以上

⁶ 追加する package によっては、ライブラリの作成が不要となる場合がある。各 package のライブラリ作成方法詳細は、`/lib/meam/README` (meam の場合) 参照のこと。