

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-29Mar19.tar.gz	29-Mar-2019 16:34 134M GZIP compressed docume>
	[] lammps-30Apr19.tar.gz	03-May-2019 10:40 135M GZIP compressed docume>
	[] lammps-30Jul16.tar.gz	30-Jul-2016 12:42 90M GZIP compressed docume>
	[] lammps-30Mar18.tar.gz	30-Mar-2018 14:39 131M GZIP compressed docume>
	[] lammps-30Oct14.tar.gz	11-Nov-2014 08:15 57M GZIP 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_ip_linux.html (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
```

- ⑨ Linux サーバにログインし直し、以下のコマンドを実行してインストールした LAMMPS が表示されることを確認する。

```
$ which lmp_serial
```

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。

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スキップ) ⁴

- ① MPICH サイトにアクセスし、“ Older releases are available here.” をクリックする。

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




Downloads

MPICH is distributed under a [BSD-like license](#). NOTE: MPICH binary packages are available in many UNIX distributions and for Windows. For example, you can search for it using “yum” (on Fedora), “apt” (Debian/Ubuntu), “pkg_add” (FreeBSD) or “port”/“brew” (Mac OS). If available for your platform, this is likely the easiest installation method since it automatically checks for dependency packages and installs them. Otherwise you can use the [installation guide](#) for installing MPICH from the source code below.

Release	Platform	Download	Size
mpich-3.3.2 (stable release)	MPICH	[http]	26 MB
hydra-3.3.2 (stable release)	Hydra (mpiexec)	[http]	4 MB
mpich-3.4a2 (alpha release)	MPICH	[http]	26 MB
hydra-3.4a2 (alpha release)	Hydra (mpiexec)	[http]	4 MB

Older releases are available [here](#). Snapshots are available [here](#). MPE releases are available [here](#).

- ② 3.1.4 をクリックする。

 3.1.3/	13-NOV-2016 10:20	-
 3.1.4/	13-Nov-2016 10:20	-
 3.1/	13-Nov-2016 10:20	-

- ③ mpich-3.1.4 をダウンロードする。

 Parent Directory	-
 hydra-3.1.4.tar.gz	23-Feb-2015 10:08 2.9M
 mpich-3.1.4-README.txt	23-Feb-2015 10:08 32K
 mpich-3.1.4-installguide.pdf	23-Feb-2015 10:08 77K
 mpich-3.1.4-userguide.pdf	23-Feb-2015 10:08 78K
 mpich-3.1.4.tar.gz	23-Feb-2015 10:08 11M

- ④ 保存した mpich-3.1.4.tar.gz 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に置く。

- ⑤ 圧縮ファイルを解凍する。

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

- ⑥ 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
```

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が、問題が起こる場合は mpich-3.1.4.tar.gz をダウンロードする。

```
$ make install
```

- ⑦ ~/.bashrc に次の一行を追加する。

```
export PATH=${HOME}/mpich-install/bin:$PATH
```

- ⑧ Linux サーバにログインし直し、以下のコマンドを実行してインストールした MPICH が表示されることを確認する。

```
$ which mpirun
```

- ⑨ LAMMPS の並列実行モジュールをビルドする

```
$ cd ~/lammeps-30Jul16/src
```

```
$ make mpi
```

- ⑩ 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。

```
 :
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 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
```

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。

- ⑪ Linux サーバにログインし直し、以下のコマンドを実行してインストールした LAMMPS が表示されることを確認する。

```
$ which lmp_mpi
```

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。

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

- ① Winmostar を起動しファイル | 新規ファイルをクリックする。
- ② ファイル | インポート | Samples ファイル | ch4_vapor_am1bcc_dreiding_lammps.wmm をクリックする。
- ③ 破棄して読み込みをクリックする。
- ④ MD | LAMMPS | キーワード設定をクリックする。
- ⑤ LAMMPS Keyword Setup ウィンドウ右下の OK をクリックする。
- ⑥ 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 をクリックしてファイルを取得するところまで実行する。
- ⑦ MD | LAMMPS | ログを表示をクリックし、デフォルトで選択されたファイルを開く。LAMMPS が正常に実行された場合は、開かれたテキストファイルの最後の方に下図のように計算の統計情報が表示される。

```
methane.log - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
Force two-norm initial, final = 5.1726 0.116752
Force max component initial, final = 2.57702 0.0846891
Final line search alpha, max atom move = 0.0938149 0.0079451
Iterations, force evaluations = 4 15

MPI task timing breakdown:
Section | min time | avg time | max time | %varavg | %total
-----|-----|-----|-----|-----|-----
Pair    | 0         | 0         | 0         | 0.0      | 0.00
Bond    | 0         | 0         | 0         | 0.0      | 0.00
Kspace  | 0         | 0         | 0         | 0.0      | 0.00
Neigh   | 0         | 0         | 0         | 0.0      | 0.00
Comm    | 0         | 0         | 0         | 0.0      | 0.00
Output  | 0         | 0         | 0         | 0.0      | 0.00
Modify  | 0         | 0         | 0         | 0.0      | 0.00
Other   | 0         | 0         | 0         | 0.0      | 0.00

Nlocal: 5 ave 5 max 5 min
Histogram: 1 0 0 0 0 0 0 0 0 0
Nghost: 3 ave 3 max 3 min
Histogram: 1 0 0 0 0 0 0 0 0 0
Neighs: 10 ave 10 max 10 min
Histogram: 1 0 0 0 0 0 0 0 0 0

Total # of neighbors = 10
Ave neighs/atom = 2
Ave special neighs/atom = 4
Neighbor list builds = 0
Dangerous builds = 0
System init for write_restart ...
PPPM initialization ...
G vector (1/distance) = 0.130007
grid = 3 3 3
stencil order = 4
estimated absolute RMS force accuracy = 0.00266588
estimated relative force accuracy = 8.02823e-006
using double precision FFTs
3d grid and FFT values/proc = 512 27
Total wall time: 0:00:00
.
```

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 ~/lammmps-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。