Installation and Configuration of LAMMPS Using CMake On windows-linux subsystem

1. Install Linux on Windows (WSL)

To run Ubuntu on Windows, you need to install Windows Subsystem for Linux (WSL). Follow these steps:

1. Open PowerShell as Administrator.
2. Run the following command to enable WSL (two times the same command or unless installed):

```
wsl --install -d ubuntu
```

3. Once the installation is complete, restart your computer.
4. After restarting, you can open Ubuntu from the Start menu.
5. Follow the prompts to complete the installation and set up your username and password.

Once WSL is installed, you will have access to a Linux terminal on your Windows machine.

Under the network section, in the left påne of the folders UI (where you get after clicking **This PC** button), you will have the access to the ubuntu folder. Within that folder you will get the home folder, which you can enable for the quick access.

Within the home folder make a directory named **Document**, where you will download your lammps using github cloaning.

2. Prerequisites

Before starting the installation process, ensure the following packages are installed on your UBUNTU sub-system. Use the following command in ubuntu terminal to install them:

```
sudo apt update
sudo apt install -y build-essential cmake g++ gfortran git
sudo apt install libfftw3-dev libjpeg-dev libpng-dev
sudo apt install libhdf5-dev libopenmpi-dev
sudo apt install liblapack-dev libblas-dev
sudo apt install libpython3-dev libcurl4-openssl-dev
sudo apt install libzstd-dev gedit
sudo apt install ffmpeg
sudo apt install pkg-config
sudo apt install libnetcdf-dev
sudo apt install libpnetcdf-dev
```

Some other packages are given as:

```
sudo apt install cmake-curses-gui
sudo apt-get install libkim-api-dev
```

Note: After copying the command from the PDF instruction file, Make sure that the extra spaces and other discrepancies are removed before running commands for the installation of the files.

Common Issues During Configuration

- Missing libraries such as `libkim-api.so` or `libhdf5`. Make sure these libraries are also installed. - make sure that the other few supportive libraries like .

Note: This list covers common dependencies needed for LAMMPS. If you enable additional packages (e.g., KIM or Voronoi), ensure their dependencies are installed. You can install these easily by using `sudo apt install package_name` command.

3. Download and Setup LAMMPS

Clone the LAMMPS repository and prepare the build directory:

```
# Clone the LAMMPS repository
git clone https://github.com/lammps/lammps.git
cd lammps

# Create a build directory
mkdir build
cd build
```

4. Configure LAMMPS with CMake

While in the build directory, run cmake:

```
ccmake ../cmake
```

This command will configure LAMMPS by copying the cmake.txt file from the cmake directory within the downloaded LAMMPS folder.

Press the **"c"** button to configure, then press **"e"** to remake the configuration and click enter on the packages you want to install. Once all necessary packages are selected, press **"c"** to finalize the configuration.

Note: Please avoid enabling every module package for installation. Only enable the packages listed below. And also please do not alter the status of other packages.

Required LAMMPS Packages

ASPHERE ATC AWPMD BOCS BODY CG-DNA CG-SPICA CLASS2 COLLOID COMPRESS CORESHELL DI-ELECTRIC DIFFRACTION DIPOLE DPD-BASIC DPD-MESO DPD-REACT DPD-SMOOTH DRUDE EFF ELECTRODE EXTRA-PAIR FEP GRANULAR H5MD KIM KSPACE LAT-BOLTZ MANIFOLD MANYBODY MC MEAM MGPT MISC MOFFF MOLECULE MOLFILE NETCDF OPENMP OPT ORIENT PERI PHONON POEMS PYTHON QEQ

QTB REPLICA RIGID SHOCK SMTBQ SPH SRD TALLY UEF VORONOI

The 'CMAKE_INSTALL_PREFIX' specifies where LAMMPS will be installed. If the configuration fails, ensure that all required dependencies (see Section 1) are installed.

5. Build and Install LAMMPS

Compile and install LAMMPS:

```
make -j$(nproc)    # Compile using all available cores
make install       # Install to the specified directory
```

6. Make LAMMPS Executable Globally Available

To make the LAMMPS executable accessible from any directory:

1. Add the installation directory to your `PATH` variable. Edit your `~/.bashrc` file:

   ```
   nano ~/.bashrc
   ```

   You can also use **gedit** command to open which would be easier to edit and save due to the user interface.

   ```
   gedit ~/.bashrc
   ```

   Search for the folder within the lammps directory where the **lmp** file is saved after installation. After opening the **bashsrc** file please paste the path to the folder where the **lmp** is saved which looks something like this.

2. Add the following line at the end of the file:

   ```
   export PATH=~/.local/bin:$PATH
   ```

   Caution: Please replace "yourcomputername" with your own username.

3. Save and exit the editor. Then reload the `.bashrc` file:

   ```
   source ~/.bashrc
   ```

Test the Setup: Run the following command to verify:

```
lmp -h
```

This should display the help information for the LAMMPS executable.

7. Summary

This guide provides a step-by-step process to install LAMMPS using CMake, resolve common issues, and configure the environment for ease of use. If you encounter additional issues, refer to the LAMMPS documentation or community forums for support.

8. In case of failure

Take help from the **C H A T G P T** (free version can make it through). Follow step by step and carefully read the error message. Do not give up until installed and executable globally (from any directory) within the home environment.