

# CP2K 安装指南使用和数学库 替换（8 核 32G） 实验手册



华为技术有限公司



# 目录

<b>1 CP2K 安装指南和数学库替换（8 核 32G）</b>	<b>2</b>
1.1 实验介绍	2
1.1.1 关于本实验	2
1.1.2 实验目的	2
1.1.3 实验环境说明	2
1.1.4 软件介绍	2
1.2 配置编译环境	3
1.2.1 下载安装包	3
1.2.2 安装 gmp	4
1.2.3 安装 mpfr	4
1.2.4 安装 mpc	4
1.2.5 安装 GNU	5
1.2.6 安装 openmpi	5
1.2.7 安装 cmake	6
1.2.8 安装 boost	7
1.2.9 安装 libint	7
1.2.10 安装 fftw	8
1.2.11 安装 lapack	8
1.2.12 安装 scalapack	8
1.2.13 安装 elpa	9
1.2.14 安装 spglib	9
1.2.15 安装 libxc	10
1.2.16 安装 gsl	10
1.2.17 安装 plumed	10
1.3 编译和安装 CP2K	11
1.4 运行 CP2K 测试算例	12
1.5 替换为 BLAS 库再次运行 CP2K 测试算例	13
1.6 测试结果对比	15
1.7 思考题	15

# 1

## CP2K 安装指南和数学库替换（8 核 32G）

### 1.1 实验介绍

#### 1.1.1 关于本实验

本实验旨在体验鲲鹏加速库的使用方法。并以数学库为例，展示了数学库对 CP2K 性能带来的提升。

#### 1.1.2 实验目的

- 掌握 CP2K 软件的安装使用。
- 掌握鲲鹏数学库在 CP2K 中的使用。

#### 1.1.3 实验环境说明

本实验在华为云上完成，推荐配置如下：

表1-1

ECS名称	规格
ecs-kp-test	鲲鹏计算   鲲鹏通用计算增强型   kc1.2xlarge.4   8核   32GB;openEuler   openEuler 20.03 64bit with ARM;高IO   40GB; 全动态BGP   独享   按流量计费   5Mbit/s;

#### 1.1.4 软件介绍

本实验涉及的软件及连接如下表，请提前准备好相关软件。（为方便实验及节约时间，本次实验将直接提供已下载的安装包）

表1-2 软件链接

软件名称	版本	下载链接
CP2K	7.1	<a href="https://github.com/cp2k/cp2k/archive/v7.1.0.tar.gz">https://github.com/cp2k/cp2k/archive/v7.1.0.tar.gz</a>
gmp	6.1.0	<a href="http://gcc.gnu.org/pub/gcc/infrastructure/">http://gcc.gnu.org/pub/gcc/infrastructure/</a>

mpfr	3.1.4	
mpc	1.0.3	
GNU	9.1.0	<a href="https://ftp.gnu.org/gnu/gcc/gcc-9.1.0/">https://ftp.gnu.org/gnu/gcc/gcc-9.1.0/</a>
openmpi	4.0.1	<a href="https://download.open-mpi.org/release/open-mpi/v4.0/openmpi-4.0.1.tar.gz">https://download.open-mpi.org/release/open-mpi/v4.0/openmpi-4.0.1.tar.gz</a>
libint	2.6.0	<a href="https://github.com/evaleev/libint/archive/v2.6.0.tar.gz">https://github.com/evaleev/libint/archive/v2.6.0.tar.gz</a>
libxc	4.3.4	<a href="http://forge.abinit.org/fallbacks/libxc-4.3.4.tar.gz">http://forge.abinit.org/fallbacks/libxc-4.3.4.tar.gz</a>
gftw	3.3.8	<a href="https://www.cp2k.org/static/downloads/fftw-3.3.8.tar.gz">https://www.cp2k.org/static/downloads/fftw-3.3.8.tar.gz</a>
lapack	3.8.0	<a href="https://www.cp2k.org/static/downloads/lapack-3.8.0.tgz">https://www.cp2k.org/static/downloads/lapack-3.8.0.tgz</a>
scalapack	2.1.0	<a href="https://www.cp2k.org/static/downloads/scalapack-2.1.0.tgz">https://www.cp2k.org/static/downloads/scalapack-2.1.0.tgz</a>
cmake	3.16.4	<a href="https://cmake.org/files/v3.16/cmake-3.16.4.tar.gz">https://cmake.org/files/v3.16/cmake-3.16.4.tar.gz</a>
boost	1.72	<a href="https://boostorg.jfrog.io/artifactory/main/release/1.72.0/source/boost_1_72_0.tar.gz">https://boostorg.jfrog.io/artifactory/main/release/1.72.0/source/boost_1_72_0.tar.gz</a>
dbcsr	2.0.1	<a href="https://github.com/cp2k/dbcsr/releases/download/v2.0.1/dbcsr-2.0.1.tar.gz">https://github.com/cp2k/dbcsr/releases/download/v2.0.1/dbcsr-2.0.1.tar.gz</a>
elpa	2019.05.001	<a href="https://www.cp2k.org/static/downloads/elpa-2019.05.001.tar.gz">https://www.cp2k.org/static/downloads/elpa-2019.05.001.tar.gz</a>
spglib	1.12.2	<a href="https://github.com/spglib/spglib/archive/v1.11.2.1.tar.gz">https://github.com/spglib/spglib/archive/v1.11.2.1.tar.gz</a>
gsl	2.6	<a href="http://mirrors.ustc.edu.cn/gnu/gsl/gsl-2.6.tar.gz">http://mirrors.ustc.edu.cn/gnu/gsl/gsl-2.6.tar.gz</a>
plumed	2.5.2	<a href="https://www.cp2k.org/static/downloads/plumed-2.5.2.tgz">https://www.cp2k.org/static/downloads/plumed-2.5.2.tgz</a>
测试算例	H2O-64.inp	软件自带测试算例

## 1.2 配置编译环境

### 1.2.1 下载安装包

此实验所需的安装包已在实验一中进行了下载，在 home 目录下的 cp2k 项目中。

将安装包复制到/home 目录下：

```
[root@ecs-kp-test ~]# cd /home
[root@ecs-kp-test home]# cp -r cp2k/TarsForInstall/. /home/
```

## 1.2.2 安装 gmp

步骤 1 执行以下命令使用 yum 安装依赖包。

```
[root@ecs-kp-test home]# yum install --nogpgcheck -y m4
```

步骤 2 执行以下命令解压安装包并进入解压文件路径。

```
[root@ecs-kp-test home]# tar -xvf gmp-6.1.0.tar.bz2
[root@ecs-kp-test home]# cd gmp-6.1.0
```

步骤 3 执行以下命令进行编译安装。

```
[root@ecs-kp-test gmp-6.1.0]# ./configure --prefix=/path/to/GMP/
[root@ecs-kp-test gmp-6.1.0]# make -j
[root@ecs-kp-test gmp-6.1.0]# make install
```

步骤 4 执行以下命令加载环境变量。

```
[root@ecs-kp-test gmp-6.1.0]# export LD_LIBRARY_PATH=/path/to/GMP/lib:$LD_LIBRARY_PATH
```

## 1.2.3 安装 mpfr

步骤 1 执行以下命令解压安装包并进入解压文件路径。

```
[root@ecs-kp-test gmp-6.1.0]# cd /home
[root@ecs-kp-test home]# tar -xvf mpfr-3.1.4.tar.bz2
[root@ecs-kp-test home]# cd mpfr-3.1.4
```

步骤 2 执行以下命令进行编译安装。

```
[root@ecs-kp-test mpfr-3.1.4]# ./configure --prefix=/path/to/MPFR --with-gmp=/path/to/GMP
[root@ecs-kp-test mpfr-3.1.4]# make -j
[root@ecs-kp-test mpfr-3.1.4]# make install
```

步骤 3 执行以下命令加载环境变量。

```
[root@ecs-kp-test mpfr-3.1.4]# export LD_LIBRARY_PATH=/path/to/MPFR/lib:$LD_LIBRARY_PATH
```

## 1.2.4 安装 mpc

步骤 1 执行以下命令解压安装包并进入解压文件路径。

```
[root@ecs-kp-test mpfr-3.1.4]# cd /home
[root@ecs-kp-test home]# tar -zxvf mpc-1.0.3.tar.gz
[root@ecs-kp-test home]# cd mpc-1.0.3
```

步骤 2 执行以下命令进行编译安装。

```
[root@ecs-kp-test mpc-1.0.3]# ./configure --prefix=/path/to/MPC --with-gmp=/path/to/GMP --with-mpfr=/path/to/MPFR
```

```
[root@ecs-kp-test mpc-1.0.3]# make -j
[root@ecs-kp-test mpc-1.0.3]# make install
```

步骤 3 执行以下命令加载环境变量。

```
[root@ecs-kp-test mpc-1.0.3]# export LD_LIBRARY_PATH=/path/to/MPC/lib:$LD_LIBRARY_PATH
```

## 1.2.5 安装 GNU

步骤 1 执行以下命令解压安装包并进入解压文件路径。

```
[root@ecs-kp-test mpc-1.0.3]# cd /home
[root@ecs-kp-test home]# tar -vxf gcc-9.1.0.tar.xz
[root@ecs-kp-test home]# cd gcc-9.1.0
```

步骤 2 执行以下命令创建 obj 文件并进入。

```
[root@ecs-kp-test gcc-9.1.0]# mkdir obj
[root@ecs-kp-test gcc-9.1.0]# cd obj
```

步骤 3 执行以下命令进行编译安装。

```
[root@ecs-kp-test obj]# ./configure --disable-multilib --enable-languages="c,c++,fortran" --
prefix=/path/to/GNU --disable-static --enable-shared --with-gmp=/path/to/GMP --with-
mpfr=/path/to/MPFR --with-mpc=/path/to/MPC
[root@ecs-kp-test obj]# make -j
[root@ecs-kp-test obj]# make install
```

步骤 4 设置默认 gcc 版本为 9.1.0。

```
[root@ecs-kp-test obj]# mkdir -p /usr/bin/gcc-7.3.0
[root@ecs-kp-test obj]# mv /usr/bin/gcc /usr/bin/gcc-7.3.0
[root@ecs-kp-test obj]# ln -s /path/to/GNU/bin/gcc /usr/bin/gcc
```

查看 gcc 版本，为 9.1.0 版本则设置成功：

```
[root@ecs-kp-test obj]# gcc -v
```

```
[root@ecs-kp-test to]# gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/path/to/GNU/libexec/gcc/aarch64-unknown-linux-gnu/9.1.0/lto-wrapper
Target: aarch64-unknown-linux-gnu
Configured with: ../configure --disable-multilib --enable-languages=c,c++,fortran --prefix=/path/to/GNU --disable-static --enable-
shared --with-gmp=/path/to/GMP --with-mpfr=/path/to/MPFR --with-mpc=/path/to/MPC
Thread model: posix
gcc version 9.1.0 (GCC)
```

步骤 5 执行以下命令加载环境变量。

```
[root@ecs-kp-test obj]# export PATH=/path/to/GNU/bin:$PATH
[root@ecs-kp-test obj]# export LD_LIBRARY_PATH=/path/to/GNU/lib64:$LD_LIBRARY_PATH
```

## 1.2.6 安装 openmpi

步骤 1 执行以下命令使用 yum 安装依赖包。

```
[root@ecs-kp-test obj]# cd /home
[root@ecs-kp-test home]# yum install --nogpgcheck numactl-devel-* systemd-devel-* gcc-gfortran
```

步骤 2 执行以下命令解压安装包并进入解压文件路径。

```
[root@ecs-kp-test home]# tar -xvf openmpi-4.0.1.tar.gz
[root@ecs-kp-test home]# cd openmpi-4.0.1
```

步骤 3 执行以下命令进行编译安装。

```
[root@ecs-kp-test openmpi-4.0.1]# ./configure --prefix=/path/to/OPENMPI --enable-pretty-print-stacktrace --enable-orterun-prefix-by-default --enable-mpi1-compatibility CC=gcc CXX=g++ FC=gfortran
[root@ecs-kp-test openmpi-4.0.1]# make -j
[root@ecs-kp-test openmpi-4.0.1]# make install
```

步骤 4 执行以下命令加载环境变量。

```
[root@ecs-kp-test openmpi-4.0.1]# export PATH=/path/to/OPENMPI/bin:$PATH
```

步骤 5 验证 openmpi 是否安装成功。

```
which mpirun
```

```
[root@ecs-kp-test ~]# which mpirun
/path/to/OPENMPI/bin/mpirun
[root@ecs-kp-test ~]#
```

## 1.2.7 安装 cmake

步骤 1 执行以下命令使用 yum 安装依赖包。

```
[root@ecs-kp-test openmpi-4.0.1]# cd /home
[root@ecs-kp-test home]# yum -y install --nogpgcheck --nogpgcheck openssl-devel
```

步骤 2 执行以下命令解压安装包并进入解压文件路径。

```
[root@ecs-kp-test home]# tar -xvf cmake-3.16.4.tar.gz
[root@ecs-kp-test home]# cd cmake-3.16.4
```

步骤 3 执行以下命令进行编译安装。

```
[root@ecs-kp-test cmake-3.16.4]# ./configure --prefix=/path/to/CMAKE
[root@ecs-kp-test cmake-3.16.4]# make -j16&&make install
```

步骤 4 执行以下命令加载环境变量。

```
[root@ecs-kp-test cmake-3.16.4]# export PATH=/path/to/CMAKE/bin:$PATH
```

步骤 5 验证 cmake 是否安装成功

```
which cmake
```

```
[root@ecs-kp-test ~]# which cmake
/path/to/CMAKE/bin/cmake
[root@ecs-kp-test ~]#
```

## 1.2.8 安装 boost

步骤 1 执行以下命令解压安装包并进入解压文件路径。

```
[root@ecs-kp-test cmake-3.16.4]# cd /home
[root@ecs-kp-test home]# tar -xvf boost_1_72_0.tar.gz
[root@ecs-kp-test home]# cd boost_1_72_0
```

步骤 2 执行以下命令进行编译安装。

```
[root@ecs-kp-test boost_1_72_0]# ./bootstrap.sh
[root@ecs-kp-test boost_1_72_0]# ./b2 install --prefix=/path/to/BOOST
```

## 1.2.9 安装 libint

步骤 1 执行以下命令创建安装目录使用 yum 安装依赖包。

```
[root@ecs-kp-test boost_1_72_0]# cd /home
[root@ecs-kp-test home]# mkdir -p /path/to/EXTRA
[root@ecs-kp-test home]# mkdir -p /path/to/EXTRA/mathlib
[root@ecs-kp-test home]# yum -y install --nogpgcheck gmp-devel.aarch64 libudev* libtool
```

步骤 2 执行以下命令解压安装包并进入解压文件路径进行配置。

```
[root@ecs-kp-test home]# tar -xvf libint-2.6.0.tar.gz
[root@ecs-kp-test home]# cd libint-2.6.0
[root@ecs-kp-test libint-2.6.0]# ./autogen.sh
```

步骤 3 执行以下命令创建目录并进入。

```
[root@ecs-kp-test libint-2.6.0]# mkdir build
[root@ecs-kp-test libint-2.6.0]# cd build
```

步骤 4 执行以下命令加载环境变量。

```
[root@ecs-kp-test build]# export LDFLAGS="-L/path/to/GMP/lib -L/path/to/BOOST/lib"
[root@ecs-kp-test build]# export CPPFLAGS="-I/path/to/BOOST/include/ -I/path/to/GMP/include"
```

步骤 5 执行以下命令进行配置并编译。

```
[root@ecs-kp-test build]# ../configure CXX=mpicxx --enable-eri=1 --enable-eri2=1 --enable-eri3=1 --
with-max-am=4 --with-eri-max-am=4,3 --with-eri2-max-am=6,5 --with-eri3-max-am=6,5 --with-opt-
am=3 --enable-generic-code --disable-unrolling --with-libint-exportdir=libint_cp2k_lmax4
[root@ecs-kp-test build]# make export
```

步骤 6 执行以下命令解压 libint\_cp2k\_lmax4.tgz 安装包并进入。

```
[root@ecs-kp-test build]# tar -xvf libint_cp2k_lmax4.tgz
[root@ecs-kp-test build]# cd libint_cp2k_lmax4
```



步骤 7 执行以下命令进行编译安装并清除临时环境变量。

```
[root@ecs-kp-test libint_cp2k_lmax4]# ./configure --prefix=/path/to/EXTRA/libint2 CC=mpicc  
CXX=mpicxx FC=mpifort --enable-fortran --enable-shared  
[root@ecs-kp-test libint_cp2k_lmax4]# make -j  
[root@ecs-kp-test libint_cp2k_lmax4]# make install  
[root@ecs-kp-test libint_cp2k_lmax4]# unset LDFlags  
[root@ecs-kp-test libint_cp2k_lmax4]# unset CPPFlags
```

## 1.2.10 安装 fftw

步骤 1 执行以下命令解压安装包并进入解压文件路径。

```
[root@ecs-kp-test libint_cp2k_lmax4]# cd /home  
[root@ecs-kp-test home]# tar -zxvf fftw-3.3.8.tar.gz  
[root@ecs-kp-test home]# cd fftw-3.3.8
```

步骤 2 执行以下命令进行编译安装。

```
[root@ecs-kp-test fftw-3.3.8]# ./configure CC=gcc F77=gfortran --enable-shared --enable-threads --  
enable-openmp --enable-mpi MPICC=mpicc --prefix=/path/to/EXTRA/fftw3  
[root@ecs-kp-test fftw-3.3.8]# make -j4  
[root@ecs-kp-test fftw-3.3.8]# make install
```

## 1.2.11 安装 lapack

步骤 1 执行以下命令解压安装包并进入解压文件路径。

```
[root@ecs-kp-test fftw-3.3.8]# cd /home  
[root@ecs-kp-test home]# tar -xvf lapack-3.8.0.tgz  
[root@ecs-kp-test home]# cd lapack-3.8.0
```

步骤 2 执行以下命令生成“make.inc”文件并进行编译。

```
[root@ecs-kp-test lapack-3.8.0]# cp make.inc.example make.inc  
[root@ecs-kp-test lapack-3.8.0]# make -j
```

此处若出现测试错误，可使用下列命令：

```
[root@ecs-kp-test lapack-3.8.0]# ulimit -s unlimited
```

步骤 3 执行以下命令复制静态库到另外目录。

```
[root@ecs-kp-test lapack-3.8.0]# cp *.a /path/to/EXTRA/mathlib
```

## 1.2.12 安装 scalapack

步骤 1 执行以下命令解压安装包并进入解压文件路径。

```
[root@ecs-kp-test lapack-3.8.0]# cd /home  
[root@ecs-kp-test home]# tar -xvf scalapack-2.1.0.tgz
```

```
[root@ecs-kp-test home]# cd scalapack-2.1.0
```

步骤 2 执行以下命令生成“SLmake.inc”文件。

```
[root@ecs-kp-test scalapack-2.1.0]# cp SLmake.inc.example SLmake.inc
```

步骤 3 执行以下命令进行修改“SLmake.inc”文件。

```
[root@ecs-kp-test scalapack-2.1.0]# vim SLmake.inc
```

输入“:set nu”显示行数。

按“i”进入编辑模式，并修改 58、59 行内容为：

```
BLASLIB    = /path/to/EXTRA/mathlib/librefblas.a  
LAPACKLIB  = /path/to/EXTRA/mathlib/liblapack.a
```

```
58 BLASLIB    = /path/to/EXTRA/mathlib/librefblas.a  
59 LAPACKLIB  = /path/to/EXTRA/mathlib/liblapack.a
```

步骤 4 按“Esc”键，输入:wq!，按“Enter”保存并退出编辑。

步骤 5 执行以下命令进行编译并复制生成静态库到另外目录。

```
[root@ecs-kp-test scalapack-2.1.0]# make  
[root@ecs-kp-test scalapack-2.1.0]# cp *.a /path/to/EXTRA/mathlib
```

## 1.2.13 安装 elpa

步骤 1 执行以下命令解压安装包并进入解压文件路径。

```
[root@ecs-kp-test scalapack-2.1.0]# cd /home  
[root@ecs-kp-test home]# tar -xvf elpa-2019.05.001.tar.gz  
[root@ecs-kp-test home]# cd elpa-2019.05.001
```

步骤 2 执行以下命令进行编译安装。

```
[root@ecs-kp-test elpa-2019.05.001]# ./configure --prefix=/path/to/EXTRA/elpa --enable-openmp --  
enable-shared=no LIBS="/path/to/EXTRA/mathlib/libscalapack.a /path/to/EXTRA/mathlib/liblapack.a  
/path/to/EXTRA/mathlib/librefblas.a" --disable-ssse3 --disable-ssse3-assembly --disable-avx --disa-  
ble-avx2 --  
disable-mpi-module  
[root@ecs-kp-test elpa-2019.05.001]# make  
[root@ecs-kp-test elpa-2019.05.001]# make install
```

## 1.2.14 安装 spglib

步骤 1 执行以下命令解压安装包并进入解压文件路径。

```
[root@ecs-kp-test elpa-2019.05.001]# cd /home  
[root@ecs-kp-test home]# tar -xvf spglib-1.11.2.1.tar.gz  
[root@ecs-kp-test home]# cd spglib-1.11.2.1
```

步骤 2 创建“build”目录并进入。

```
[root@ecs-kp-test spglib-1.11.2.1]# mkdir build
[root@ecs-kp-test spglib-1.11.2.1]# cd build
```

步骤 3 执行以下命令进行编译安装。

```
[root@ecs-kp-test build]# cmake .. -DCMAKE_INSTALL_PREFIX="/path/to/EXTRA/spglib112"
[root@ecs-kp-test build]# make -j
[root@ecs-kp-test build]# make install
```

## 1.2.15 安装 libxc

步骤 1 执行以下命令解压安装包并进入解压文件路径。

```
[root@ecs-kp-test build]# cd /home
[root@ecs-kp-test home]# tar -xvf libxc-4.3.4.tar.gz
[root@ecs-kp-test home]# cd libxc-4.3.4
```

步骤 2 执行以下命令进行编译安装。

```
[root@ecs-kp-test libxc-4.3.4]# ./configure FC=gfortran CC=gcc --prefix=/path/to/EXTRA/libxc434
[root@ecs-kp-test libxc-4.3.4]# make -j
[root@ecs-kp-test libxc-4.3.4]# make install
```

## 1.2.16 安装 gsl

步骤 1 执行以下命令解压安装包并进入解压文件路径。

```
[root@ecs-kp-test libxc-4.3.4]# cd /home
[root@ecs-kp-test home]# tar -xvf gsl-2.6.tar.gz
[root@ecs-kp-test home]# cd gsl-2.6
```

步骤 2 执行以下命令进行编译安装。

```
[root@ecs-kp-test gsl-2.6]# ./configure --prefix=/path/to/EXTRA/gsl
[root@ecs-kp-test gsl-2.6]# make -j
[root@ecs-kp-test gsl-2.6]# make install
```

步骤 3 执行以下命令加载环境变量。

```
[root@ecs-kp-test gsl-2.6]# export LD_LIBRARY_PATH=/path/to/EXTRA/gsl/lib:$LD_LIBRARY_PATH
```

## 1.2.17 安装 plumed

步骤 1 执行以下命令解压安装包并进入解压文件路径。

```
[root@ecs-kp-test gsl-2.6]# cd /home
[root@ecs-kp-test home]# tar -xvf plumed-2.5.2.tgz
[root@ecs-kp-test home]# cd plumed-2.5.2
```

步骤 2 执行以下命令进行编译安装。

```
[root@ecs-kp-test plumed-2.5.2]# ./configure CXX=mpicxx CC=mpicc FC=mpifort --
prefix=/path/to/EXTRA/plumed252 --enable-external-blas --enable-gsl --enable-external-lapack
LDFLAGS=-L/path/to/EXTRA/mathlib LIBS="/path/to/EXTRA/mathlib/librefblas.a
/path/to/EXTRA/mathlib/liblapack.a"
[root@ecs-kp-test plumed-2.5.2]# make -j
[root@ecs-kp-test plumed-2.5.2]# make install
```

步骤 3 执行以下命令加载环境变量。

```
[root@ecs-kp-test plumed-2.5.2]# export
LD_LIBRARY_PATH=/path/to/EXTRA/plumed252/lib:$LD_LIBRARY_PATH
```

## 1.3 编译和安装 CP2K

步骤 1 执行以下命令解压安装包并进入解压文件路径。

```
[root@ecs-kp-test plumed-2.5.2]# cd /home
[root@ecs-kp-test home]# tar xvf cp2k-7.1.0.tar.gz
[root@ecs-kp-test home]# cd cp2k-7.1.0/arch
```

步骤 2 执行以下命令创建配置文件。

```
[root@ecs-kp-test arch]# vi Linux-GCC-gfortran.psmf
```

步骤 3 按“i”进入编辑模式，修改文件，将以下内容复制至文件中。

```
CC =mpicc
FC =mpif90
LD =mpif90
AR =ar -r
GNU_PATH = /path/to/EXTRA/
MATHLIBPATH = /path/to/EXTRA/mathlib
include $(GNU_PATH)/plumed252/lib/plumed/src/lib/Plumed.inc.static
ELPA_VER = 2019.05.001
ELPA_INC = $(GNU_PATH)/elpa/include/elpa_openmp-$(ELPA_VER)
ELPA_LIB = $(GNU_PATH)/elpa/lib
FFTW_INC = $(GNU_PATH)/fftw3/include
FFTW_LIB = $(GNU_PATH)/fftw3/lib
LIBINT_INC = $(GNU_PATH)/libint2/include
LIBINT_LIB = $(GNU_PATH)/libint2/lib
LIBXC_INC = $(GNU_PATH)/libxc434/include
LIBXC_LIB = $(GNU_PATH)/libxc434/lib
SPGLIB_INC = $(GNU_PATH)/spglib112/include
SPGLIB_LIB = $(GNU_PATH)/spglib112/lib
CFLAGS = -O2 -g -mtune=native
DFFLAGS = -D_ELPA -D_FFTW3 -D_LIBINT -D_LIBXC
DFFLAGS += -D_MPI_VERSION=3 -D_PLUMED2 -D_SPGLIB
DFFLAGS += -D_parallel -D_SCALAPACK
```

```
FCFLAGS      = $(CFLAGS) $(DFLAGS)
FCFLAGS      += -ffree-form -ffree-line-length-none
FCFLAGS      += -fopenmp
FCFLAGS      += -ftree-vectorize -funroll-loops -std=f2008
FCFLAGS      += -I$(ELPA_INC)/elpa -I$(ELPA_INC)/modules
FCFLAGS      += -I$(FFTW_INC) -I$(LIBINT_INC) -I$(LIBXC_INC)
LDFLAGS      = $(FCFLAGS)
GSLBLAS_LIB = $(GNU_PATH)/gsl/lib
PLUMED_LIB = $(GNU_PATH)/plumed252/lib
LIBS         = -L$(GSLBLAS_LIB) -L$(PLUMED_LIB) -lgsl -lgslcblas -lz -lplumed -lplumedKernel
LIBS         += $(ELPA_LIB)/libelpa_openmp.a
LIBS         += $(LIBXC_LIB)/libxcf03.a
LIBS         += $(LIBXC_LIB)/libxc.a
LIBS         += $(LIBINT_LIB)/libint2.a
LIBS         += $(SPGLIB_LIB)/libsymspg.a
LIBS         += $(FFTW_LIB)/libfftw3.a
LIBS         += $(FFTW_LIB)/libfftw3_threads.a
LIBS         += $(MATHLIBPATH)/libscalapack.a
LIBS         += $(MATHLIBPATH)/liblapack.a
LIBS         += $(MATHLIBPATH)/librefblas.a
LIBS         += -ldl -lpthread -lstdc++
```

按“Esc”键，输入:wq!，按“Enter”保存并退出编辑。

步骤 4 执行以下命令进入目录中并将 dbcsr 压缩文件复制至当前目录。

```
[root@ecs-kp-test arch]# cd ../exts/
[root@ecs-kp-test exts]# cp /home/dbcsr-2.0.1.tar.gz ./
```

步骤 5 解压并把 dbcsr-2.0.1 所有文件移到上一目录 dbcsr 中，最后删除 dbcsr 压缩包及其解压后的文件。

```
[root@ecs-kp-test exts]# tar -xvf dbcsr-2.0.1.tar.gz
[root@ecs-kp-test exts]# mv dbcsr-2.0.1/* dbcsr/
[root@ecs-kp-test exts]# rm -f dbcsr-2.0.1.tar.gz
[root@ecs-kp-test exts]# rm -rf dbcsr-2.0.1
```

步骤 6 执行以下命令进行编译安装。

```
[root@ecs-kp-test exts]# cd /home/cp2k-7.1.0
[root@ecs-kp-test cp2k-7.1.0]# make -j 16 ARCH=Linux-GCC-gfortran VERSION=psmp
```

## 1.4 运行 CP2K 测试算例

以下是在 8 核 32GB 的规格上使用 H2O-64 测试用例。如果实验时间紧张，也可用 H2O-32 测试用例。（具体差别会以红色字体在过程中标出）

步骤 1 执行以下命令加载环境变量。

```
[root@ecs-kp-test cp2k-7.1.0]# export PATH=/home/cp2k-7.1.0/exe/Linux-GCC-gfortran:$PATH
[root@ecs-kp-test cp2k-7.1.0]# export LD_LIBRARY_PATH=/path/to/EXTRA/mathlib:$LD_LIBRARY_PATH
```

步骤 2 执行以下命令取消 openEuler 操作系统的进程绑核，提高 cpu 使用率。

```
[root@ecs-kp-test cp2k-7.1.0]# export OMP_PROC_BIND=FALSE
```

步骤 3 执行以下命令运行 CP2K 自带测试算例。

```
[root@ecs-kp-test cp2k-7.1.0]# cd benchmarks/QS
[root@ecs-kp-test QS]# mpirun --allow-run-as-root -np 8 -x OMP_NUM_THREADS=1 cp2k.psmf H2O-64.inp 2>&1 | tee -a cp2k.H2O-64.inp.log
```

如果使用 H2O-32 测试算例，则命令行为：

```
[root@ecs-kp-test QS]# mpirun --allow-run-as-root -np 8 -x OMP_NUM_THREADS=1 cp2k.psmf H2O-32.inp 2>&1 | tee -a cp2k.H2O-32.inp.log
```

步骤 4 查看测试结果。

测试开始时部分截图：

```
**** ** PROGRAM STARTED AT 2022-02-26 14:43:14.362
**** ** PROGRAM STARTED ON ecs-kp-test
**** ** PROGRAM STARTED BY root
**** ** PROGRAM PROCESS ID 611829
**** ** PROGRAM STARTED IN /home/cp2k-7.1.0/benchmarks/QS
```

测试结束后部分截图：

```
-----
**** ** PROGRAM ENDED AT 2022-02-26 15:10:40.191
**** ** PROGRAM RAN ON ecs-kp-test
**** ** PROGRAM RAN BY root
**** ** PROGRAM PROCESS ID 611829
**** ** PROGRAM STOPPED IN /home/cp: Ctrl + click to follow link :/QS
[root@ecs-kp-test QS]#
```

也可在运行结束后，在当前目录下查看日志文件“cp2k.H2O-64.inp.log”：

```
[root@ecs-kp-test QS]# vi cp2k.H2O-64.inp.log
```

如果使用 H2O-32 测试算例，则命令行为：

```
[root@ecs-kp-test QS]# vi cp2k.H2O-32.inp.log
```

可以看到，完成测试的总时间为 1646 秒。

## 1.5 替换为 BLAS 库再次运行 CP2K 测试算例

步骤 1 进入/home/cp2k-7.1.0 目录并修改配置文件。

```
[root@ecs-kp-test QS]# cd /home/cp2k-7.1.0
[root@ecs-kp-test cp2k-7.1.0]# vi arch/Linux-GCC-gfortran.psmf
```

输入“:set nu”显示行数。

按 “i” 进入编辑模式，并修改 42 行内容为：

```
LIBS      +=-L/usr/local/kml/lib/kblas/nolocking/ -lkblas

40 LIBS      += $(MATHLIBPATH)/libscalapack.a
41 LIBS      += $(MATHLIBPATH)/liblapack.a
42 LIBS      +=-L/usr/local/kml/lib/kblas/nolocking/ -lkblas
```

按 “Esc” 键，输入:wq!，按 “Enter” 保存并退出编辑。

步骤 2 删除可执行程序。

```
[root@ecs-kp-test cp2k-7.1.0]# rm -rf exe
```

步骤 3 重新进行编译。

```
[root@ecs-kp-test cp2k-7.1.0]# make -j 16 ARCH=Linux-GCC-gfortran VERSION=psmp
```

步骤 4 执行以下命令运行 CP2K 自带测试算例。

```
[root@ecs-kp-test cp2k-7.1.0]# cd benchmarks/QS
[root@ecs-kp-test QS]# mpirun --allow-run-as-root -np 8 -x OMP_NUM_THREADS=1 cp2k.psmf H2O-64.inp 2>&1 | tee -a cp2k.H2O-64.inp.log
```

如果使用 H2O-32 测试算例，则命令行为：

```
[root@ecs-kp-test QS]# mpirun --allow-run-as-root -np 8 -x OMP_NUM_THREADS=1 cp2k.psmf H2O-32.inp 2>&1 | tee -a cp2k.H2O-32.inp.log
```

步骤 5 查看测试结果。

测试开始时部分截图：

```
*****
***** PROGRAM STARTED AT 2022-02-26 14:14:52.610
***** PROGRAM STARTED ON ecs-kp-test
***** PROGRAM STARTED BY root
***** PROGRAM PROCESS ID 609968
***** PROGRAM STARTED IN /home/cp2k-7.1.0/benchmarks/QS
```

测试结束后部分截图：

```
-----
*****
***** PROGRAM ENDED AT 2022-02-26 14:32:58.627
***** PROGRAM RAN ON ecs-kp-test
***** PROGRAM RAN BY root
***** PROGRAM PROCESS ID 609968
***** PROGRAM STOPPED IN /home/cp2k-7.1.0/benchmarks/QS
[root@ecs-kp-test QS]#
```

也可在运行结束后，在当前目录下查看日志文件 “cp2k.H2O-64.inp.log”：

```
[root@ecs-kp-test QS]# vi cp2k.H2O-64.inp.log
```

如果使用 H2O-32 测试算例，则命令行为：

```
[root@ecs-kp-test QS]# vi cp2k.H2O-32.inp.log
```

可以看到，完成测试的总时间为 1086 秒。

## 1.6 测试结果对比

在 8 核 32GB 的鲲鹏处理器下进行对 64H2O 算例进行测试后发现：将原先使用的库替换成鲲鹏数学库中的 BLAS 库进行计算时，时间上有大幅缩减，加速约 34%。

## 1.7 思考题

在安装 CP2K 的过程中，安装软件的顺序可以任意改变。

（答案：错误）