CP2K 安装指南使用和数学库替换（2 核 4G）实验手册

华为技术有限公司

目录

1 CP2K 安装指南和数学库替换（2 核 4G）

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.xlarge.2 | 2核 | 4GB;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	https://github.com/cp2k/cp2k/archive/v7.1.0.tar.gz
gmp	6.1.0	http://gcc.gnu.org/pub/gcc/infrastructure/

mpfr	3.1.4	
mpc	1.0.3	
GNU	9.1.0	https://ftp.gnu.org/gnu/gcc/gcc-9.1.0/
openmpi	4.0.1	https://download.open-mpi.org/release/open-mpi/v4.0/openmpi-4.0.1.tar.gz
libint	2.6.0	https://github.com/evaleev/libint/archive/v2.6.0.tar.gz
libxc	4.3.4	http://forge.abinit.org/fallbacks/libxc-4.3.4.tar.gz
gftw	3.3.8	https://www.cp2k.org/static/downloads/fftw-3.3.8.tar.gz
lapack	3.8.0	https://www.cp2k.org/static/downloads/lapack-3.8.0.tgz
scalapack	2.1.0	https://www.cp2k.org/static/downloads/scalapack-2.1.0.tgz
cmake	3.16.4	https://cmake.org/files/v3.16/cmake-3.16.4.tar.gz
boost	1.72	https://boostorg.jfrog.io/artifactory/main/release/1.72.0/source/boost_1_72_0.tar.gz
dbcsr	2.0.1	https://github.com/cp2k/dbcsr/releases/download/v2.0.1/dbcsr-2.0.1.tar.gz
elpa	2019.05.001	https://www.cp2k.org/static/downloads/elpa-2019.05.001.tar.gz
spglib	1.12.2	https://github.com/spglib/spglib/archive/v1.11.2.1.tar.gz
gsl	2.6	http://mirrors.ustc.edu.cn/gnu/gsl/gsl-2.6.tar.gz
plumed	2.5.2	https://www.cp2k.org/static/downloads/plumed-2.5.2.tgz
测试算例	H2O-32.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 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 -vxf 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 -zvxf 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

由于编译安装 GNU 比较耗时，直接提供已经编译好的压缩包。

步骤 1　进入安装路径/path/to，将下载的安装包中的 GNU 的编译结果复制到当前路径。

```
[root@ecs-kp-test mpc-1.0.3]# cd /path/to
[root@ecs-kp-test to]# cp /home/cp2k/ResOfInstall/GNU.tar.gz ./
```

步骤 2　解压并删除压缩包。

```
[root@ecs-kp-test to]# tar -zxvf GNU.tar.gz
[root@ecs-kp-test to]# rm GNU.tar.gz
```

步骤 3　设置默认 gcc 版本为 9.1.0。

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

```
[root@ecs-kp-test to]# gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/libexec/gcc/aarch64-linux-gnu/7.3.0/lto-wrapper
Target: aarch64-linux-gnu
Configured with: ../configure --prefix=/usr --mandir=/usr/share/man --infodir=/usr/share/info --enable-shared --enable-threads=pos
ix --enable-checking=release --with-system-zlib --enable-__cxa_atexit --disable-libunwind-exceptions --enable-gnu-unique-object --
enable-linker-build-id --with-linker-hash-style=gnu --enable-languages=c,c++,objc,obj-c++,fortran,lto --enable-plugin --enable-ini
tfini-array --disable-libgcj --without-isl --without-cloog --enable-gnu-indirect-function --build=aarch64-linux-gnu --with-stage1-
ldflags=' -Wl,-z,relro,-z,now' --with-boot-ldflags=' -Wl,-z,relro,-z,now' --with-multilib-list=lp64
Thread model: posix
gcc version 7.3.0 (GCC)
```

可以看到当前的 gcc 版本为 7.3.0，所以设置系统默认的 gcc 版本为此次安装的 9.1.0 版本：

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

再次查看 gcc 版本，为 9.1.0 版本则设置成功：

```
[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 4　执行以下命令加载环境变量。

```
[root@ecs-kp-test to]# export PATH=/path/to/GNU/bin:$PATH
[root@ecs-kp-test to]# export LD_LIBRARY_PATH=/path/to/GNU/lib64:$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 to]# 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 -j4
[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 install --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 -j4 && 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

由于编译安装 libint 比较耗时，直接提供已经编译好的压缩包。

步骤 1　执行以下命令创建安装目录并使用 yum 安装依赖包。

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

步骤 2　将下载的安装包中的 libint 的编译结果复制到当前路径。

```
[root@ecs-kp-test EXTRA]# cp /home/cp2k/ResOfInstall/libint2.tar.gz ./
```

步骤 3　解压并删除压缩包。

```
[root@ecs-kp-test EXTRA]# tar -zxvf libint2.tar.gz
[root@ecs-kp-test EXTRA]# rm libint2.tar.gz
```

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 EXTRA]# 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- 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-sse --disable-sse-assembly --disable-avx --disable-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

由于编译安装 libxc 比较耗时，直接提供已经编译好的压缩包。

步骤 1　进入安装目录并将下载的安装包中的 libxc 的编译结果复制到当前路径。

```
[root@ecs-kp-test build]# cd /path/to/EXTRA/
[root@ecs-kp-test EXTRA]# cp /home/cp2k/ResOfInstall/libxc434.tar.gz ./
```

步骤 2　解压并删除压缩包。

```
[root@ecs-kp-test EXTRA]# tar -zxvf libxc434.tar.gz
[root@ecs-kp-test EXTRA]# rm -rf libxc434.tar.gz
```

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 EXTRA]# 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

由于编译安装 plumed 比较耗时，直接提供已经编译好的压缩包。

步骤 1　进入安装目录并将下载的安装包中的 plumed 的编译结果复制到当前路径。

```
[root@ecs-kp-test gsl-2.6]# cd /path/to/EXTRA/
[root@ecs-kp-test EXTRA]# cp /home/cp2k/ResOfInstall/plumed252.tar.gz ./
```

步骤 2　解压并删除压缩包。

```
[root@ecs-kp-test EXTRA]# tar -zxvf plumed252.tar.gz
[root@ecs-kp-test EXTRA]# rm -rf plumed252.tar.gz
```

步骤 3　执行以下命令加载环境变量。

```
[root@ecs-kp-test EXTRA]# 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 EXTRA]# 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]# vim Linux-GCC-gfortran.psmp
```

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
DFLAGS          = -D__ELPA -D__FFTW3 -D__LIBINT -D__LIBXC
DFLAGS          += -D__MPI_VERSION=3 -D__PLUMED2 -D__SPGLIB
DFLAGS          += -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 4 ARCH=Linux-GCC-gfortran VERSION=psmp
```

1.4 运行 CP2K 测试算例

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 2 -x OMP_NUM_THREADS=1 cp2k.psmp 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 15:21:14.761
***** **  ***  *** **    PROGRAM STARTED ON                        ecs-kp-test
**    ****   ******     PROGRAM STARTED BY                               root
***** **    ** ** **    PROGRAM PROCESS ID                               9522
 **** **  *******  **   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 15:52:48.904
 ***** ** ***  *** **   PROGRAM RAN ON                              ecs-kp-test
 **    ****   ******    PROGRAM RAN BY                                     root
 ***** **    ** ** **   PROGRAM PROCESS ID                                 9522
  **** **  *******  **  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-32.inp.log”：

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

可以看到，完成测试的总时间为 1894 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]# vim arch/Linux-GCC-gfortran.psmp
```

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 4 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 2 -x OMP_NUM_THREADS=1 cp2k.psmp 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 14:47:38.792
***** **  ***   ***  **    PROGRAM STARTED ON                          ecs-kp-test
**     ****    ******      PROGRAM STARTED BY                                 root
***** **     **  **  **    PROGRAM PROCESS ID                                 7780
 ****  **   *******   **   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 15:14:54.714
***** **  ***   ***  **    PROGRAM RAN ON                              ecs-kp-test
**     ****    ******      PROGRAM RAN BY                                     root
***** **     **  **  **    PROGRAM PROCESS ID                                 7780
 ****  **   *******   **   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-32.inp.log”：

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

可以看到，完成测试的总时间为 1636 秒。

1.6 测试结果对比

在 2 核 4GB 的鲲鹏处理器下进行对 32H2O 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 13%。

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）