开源软件栈安装

- Gcc编译器（gcc、g++、gfortan）

一般在操作系统层面已提供，只需要在服务器上执行命令yum install gcc gcc-c++ gfortran即可安装

- Openmpi

可参考：https://gitee.com/openeuler/hpcrunner/blob/master/package/openmpi/4.0.1/install.sh

- Openblas：

可参考：https://gitee.com/openeuler/hpcrunner/blob/master/package/openblas/0.3.6/install.sh

```
tar -xvf openblas-v0.3.6.tar.gz
cd OpenBLAS-0.3.6
make -j
make PREFIX=/scratch/apps/TEST/yjb/PAC/install/openblas-gcc install
```

- Scalapack：

可参考：https://gitee.com/openeuler/hpcrunner/blob/master/package/scalapack/2.1.0/install.sh

```
tar -xvf scalapack-2.0.1.tgz
cd scalapack-2.0.1
cp SLmake.inc.example SLmake.inc
```

修改SLmake.inc中的58、59行，使用上述安装openblas的lib库

```
make
mkdir /scratch/apps/TEST/yjb/PAC/install/scalapack-gcc/lib
mkdir /scratch/apps/TEST/yjb/PAC/install/scalapack-gcc/include
cp libscalapack.a /scratch/apps/TEST/yjb/PAC/install/scalapack-gcc/lib
cp SRC/*.h /scratch/apps/TEST/yjb/PAC/install/scalapack-gcc/include
```

鲲鹏自研软件栈安装

- 毕昇编译器

已打包好二进制包，解压设置环境变量后直接使用，可参考
https://gitee.com/openeuler/hpcrunner/blob/master/package/bisheng/meta.sh

- 华为高性能通信库HMPI

可参考：https://gitee.com/openeuler/hpcrunner/blob/master/package/hmpi/2.3.0/install.sh

- 鲲鹏加速库KML

可参考：https://gitee.com/openeuler/hpcrunner/blob/master/package/kml/2.2.0/bisheng/install.sh

```
unzip KML_2.3.0_bisheng.zip
rpm --force --nodeps -ivh --prefix=/scratch/apps/TEST/yjb/PAC/install/kml kml-2.3.0-1.aarch64.rpm
```

基于gcc+openmpi安装qe

- 设置编译环境变量

```
module load openmpi-4.0.1
module load scalapack-2.0.1-gcc
module load openblas-0.3.6-gcc
export BLAS_LIBS="-L/scratch/apps/TEST/yjb/PAC/install/openblas-gcc/lib -lopenblas"
export LAPACK_LIBS="-L/scratch/apps/TEST/yjb/PAC/install/openblas-gcc/lib -lopenblas"
export SCALAPACK_LIBS="-L/scratch/apps/TEST/yjb/PAC/install/scalapack-gcc/lib -lscalapack"
export DEFS="__PGI"
```

- 安装qe

```
tar xf qe-6.4.1.tar.gz
cd q-e-qe-6.4.1
./configure --prefix=/scratch/apps/TEST/yjb/PAC/install/QE-gcc --enable-openmp ARCH=arm F77=gfortran
F90=gfortran FC=gfortran CC=gcc CFLAGS="-O3 -g -D__PGI -mcpu=native" FFLAGS="-O3 -g -D__PGI -mcpu=native -
fallow-argument-mismatch" LDFLAGS="$LDFLAGS" MPIF90=mpifort MPIF77=mpifort CC=mpicc
make -j64 pwall
make install
```

- 设置qe环境变量

```
export PATH=/scratch/apps/TEST/yjb/PAC/install/QE-gcc/bin:$PATH
```

基于毕昇+hmpi安装qe

- 设置编译环境变量

```
module load bisheng-3.2.0
module load hmpi-2.3.0
module load scalapack-2.0.1-bs
module load openblas-0.3.6-bs
export BLAS_LIBS="-L/scratch/apps/TEST/yjb/PAC/install/openblas-bs/lib -lopenblas"
export LAPACK_LIBS="-L/scratch/apps/TEST/yjb/PAC/install/openblas-bs/lib -lopenblas"
export SCALAPACK_LIBS="-L/scratch/apps/TEST/yjb/PAC/install/scalapack-bs/lib -lscalapack"
export DEFS="__PGI"
```

- 安装qe

```
tar xf qe-6.4.1.tar.gz
cd q-e-qe-6.4.1
./configure --prefix=/scratch/apps/TEST/yjb/PAC/install/QE-bs --enable-openmp ARCH=arm F77=flang F90=flang
FC=flang CC=clang CFLAGS="-O3 -g -D__PGI -mcpu=native" FFLAGS="-O3 -g -D__PGI -mcpu=native" LDFLAGS="$LDFLAGS"
MPIF90=mpifort MPIF77=mpifort CC=mpicc
make -j64 pwall
make install
```

- 设置qe环境变量

```
export PATH=/scratch/apps/TEST/yjb/PAC/install/QE-bs/bin:$PATH
```

基于毕昇+hmpi+kml安装qe

- 设置编译环境变量

```
module load bisheng-3.2.0
module load hmpi-2.3.0
module load kml-2.3.0
export BLAS_LIBS="-L/scratch/apps/TEST/yjb/PAC/install/kml/kml/lib/kblas/omp -lkblas"
export LAPACK_LIBS="-L/scratch/apps/TEST/yjb/PAC/install/kml/kml/lib -lklapack_full"
export SCALAPACK_LIBS="-L/scratch/apps/TEST/yjb/PAC/install/kml/kml/lib -lkscalapack_full"
export DEFS="__PGI"
```

- 安装qe

```
tar xf qe-6.4.1.tar.gz
cd q-e-qe-6.4.1
./configure --prefix=/scratch/apps/TEST/yjb/PAC/install/QE-bs-kml --enable-openmp ARCH=arm F77=flang F90=flang
FC=flang CC=clang CFLAGS="-O3 -g -D__PGI -mcpu=native" FFLAGS="-O3 -g -D__PGI -mcpu=native" LDFLAGS="$LDFLAGS"
MPIF90=mpifort MPIF77=mpifort CC=mpicc
make -j64 pwall
make install
```

- 设置qe环境变量

```
export PATH=/scratch/apps/TEST/yjb/PAC/install/QE-bs-kml/bin:$PATH
```

基于毕昇+hmpi+kml+优化选项安装qe

- 设置编译环境变量

```
module load bisheng-3.2.0
module load hmpi-2.3.0
module load kml-2.3.0
export BLAS_LIBS="-L/scratch/apps/TEST/yjb/PAC/install/kml/kml/lib/kblas/omp -lkblas"
export LAPACK_LIBS="-L/scratch/apps/TEST/yjb/PAC/install/kml/kml/lib -lklapack_full"
export SCALAPACK_LIBS="-L/scratch/apps/TEST/yjb/PAC/install/kml/kml/lib -lkscalapack_full"
export DEFS="__PGI"
```

- 安装qe

```
tar xf qe-6.4.1.tar.gz
cd q-e-qe-6.4.1
./configure --prefix=/scratch/apps/TEST/yjb/PAC/install/QE-bs-kml-xuanxiang --enable-openmp ARCH=arm F77=flang
F90=flang FC=flang CC=clang CFLAGS="-O3 -g -D__PGI -march=armv8+sve -mcpu=native -mllvm -force-customized-
pipeline=true -fuse-ld=lld" FFLAGS="-O3 -g -D__PGI -march=armv8+sve -mcpu=native -mllvm -force-customized-
pipeline=true -fuse-ld=lld" LDFLAGS="$LDFLAGS" MPIF90=mpifort MPIF77=mpifort CC=mpicc
make -j64 pwall
make install
```

- 设置qe环境变量

```
export PATH=/scratch/apps/TEST/yjb/PAC/install/QE-bs-kml-xuanxiang/bin:$PATH
```

测试算例获取

- 下载算例

```
git clone https://github.com/QEF/benchmarks.git
cd benchmarks/small-benchmarks
```

在pwscf_bench.sh脚本中提供4个小算例，本次测试以第4个算例test_4.in举例

截取pwscf_bench.sh脚本的第4个算例

```
sed -n '357,419p' pwscf_bench.sh > test_4.sh
export PSEUDO_DIR=`pwd`/pseudopotentials.d
export SCRATCH_DIR=`pwd`/output
source test_4.sh
```

基于gcc+openmpi运行qe

- 设置编译环境变量

```
module load openmpi-4.0.1
module load scalapack-2.0.1-gcc
module load openblas-0.3.6-gcc
```

- 设置qe环境变量

```
export PATH=/scratch/apps/TEST/yjb/PAC/install/QE-gcc/bin:$PATH
```

- 运行qe

```
cd benchmarks/small-benchmarks
mpirun --allow-run-as-root -np 64 -x OMP_NUM_THREADS=1 --mca btl ^openib pw.x --mca
opal_common_ucx_opal_mem_hooks 1 -input ./test_4.in 2>&1 | tee -a qe-64-gcc.log
```

运行完成后，在当前目录下生成日志qe-64-gcc.log，查看日志结尾处PWSCF这一行中WALL的数值，单位为秒，数值越低性能越优

基于毕昇+hmpi运行qe

- 设置编译环境变量

```
module load bisheng-3.2.0
module load hmpi-2.3.0
module load scalapack-2.0.1-bs
module load openblas-0.3.6-bs
```

- 设置qe环境变量

```
export PATH=/scratch/apps/TEST/yjb/PAC/install/QE-bs/bin:$PATH
```

- 运行qe

```
cd benchmarks/small-benchmarks
mpirun --allow-run-as-root -np 64 -x OMP_NUM_THREADS=1 --mca btl ^openib -x UCX_WARN_UNUSED_ENV_VARS=n pw.x -input ./test_4.in 2>&1 | tee -a qe-64-bs.log
```

运行完成后，在当前目录下生成日志qe-64-bs.log，查看日志结尾处PWSCF这一行中WALL的数值，单位为秒，数值越低性能越优

基于毕昇+hmpi+kml运行qe

- 设置编译环境变量

```
module load bisheng-3.2.0
module load hmpi-2.3.0
module load kml-2.3.0
```

- 设置qe环境变量

```
export PATH=/scratch/apps/TEST/yjb/PAC/install/QE-bs-kml/bin:$PATH
```

- 运行qe

```
cd benchmarks/small-benchmarks
mpirun --allow-run-as-root -np 64 -x OMP_NUM_THREADS=1 --mca btl ^openib -x UCX_WARN_UNUSED_ENV_VARS=n pw.x -input ./test_4.in 2>&1 | tee -a qe-64-bs-kml.log
```

运行完成后，在当前目录下生成日志qe-64-bs-kml.log，查看日志结尾处PWSCF这一行中WALL的数值，单位为秒，数值越低性能越优

基于毕昇+hmpi+kml+优化选项运行qe

- 设置编译环境变量

```
module load bisheng-3.2.0
module load hmpi-2.3.0
module load kml-2.3.0
```

- 设置qe环境变量

```
export PATH=/scratch/apps/TEST/yjb/PAC/install/QE-bs-kml-xuanxiang/bin:$PATH
```

- 运行qe

```
cd benchmarks/small-benchmarks
mpirun --allow-run-as-root -np 64 -x OMP_NUM_THREADS=1 --mca btl ^openib -x UCX_WARN_UNUSED_ENV_VARS=n pw.x -input ./test_4.in 2>&1 | tee -a qe-64-bs-kml-xuanxiang.log
```

运行完成后，在当前目录下生成日志qe-64-bs-kml-xuanxiang.log，查看日志结尾处PWSCF这一行中WALL的数值，单位为秒，数值越低性能越优

基于毕昇+hmpi+kml+优化选项+调整核数运行qe

- 设置编译环境变量

```
module load bisheng-3.2.0
module load hmpi-2.3.0
module load kml-2.3.0
```

- 设置qe环境变量

```
export PATH=/scratch/apps/TEST/yjb/PAC/install/QE-bs-kml-xuanxiang/bin:$PATH
```

- 运行qe

```
cd benchmarks/small-benchmarks
mpirun --allow-run-as-root -np 80 -x OMP_NUM_THREADS=1 --mca btl ^openib -x UCX_WARN_UNUSED_ENV_VARS=n pw.x -input ./test_4.in 2>&1 | tee -a qe-80-bs-kml-xuanxiang.log
```

运行完成后，在当前目录下生成日志qe-80-bs-kml-xuanxiang.log，查看日志结尾处PWSCF这一行中WALL的数值，单位为秒，数值越低性能越优

性能对比

编译软件栈	核数	运行时长	调优手段
开源gcc编译器+openmpi	64	58.35s	基础
毕昇编译器+hmpi	64	48.90s	使能鲲鹏自研软件栈
毕昇编译器+hmpi+鲲鹏加速库	64	44.57s	使能鲲鹏加速库
毕昇编译器+hmpi+鲲鹏加速库+芯片优化选项	64	43.85s	使能优化选项
毕昇编译器+hmpi+鲲鹏加速库+芯片优化选项	80	36.65s	调整鲲鹏多核优势