

实验8：HPL测试

1. 实验目的

- (1) 了解HPL测试的应用场景、原理
- (2) 掌握CPU峰值浮点性能的计算
- (3) 掌握HPL的单机测试与优化

2. 实验内容

- (1) 安装运行HPL性能测试

选做：

- 1. 替换不同版本的MPI框架、BLAS库进行对比

3. HPL安装运行

3.1 HPL安装：openblas和MPICH

HPL依赖BLAS库和MPI框架，这两个依赖前面的实验已经有安装，openblas和MPICH，无需重复安装。如需要替换成其他开源版本，则相关路径做对应的替换。

执行以下命令：完成hpl压缩包的下载和解压，最终解压到~/hpl路径下。

```
wget https://netlib.org/benchmark/hpl/hpl-2.3.tar.gz
gunzip hpl-2.3.tar.gz
tar xvf hpl-2.3.tar
rm hpl-2.3.tar
mv hpl-2.3 ~/hpl
```

生成Make.linux配置文件：

```
cd ~/hpl/setup
sh make_generic
cp Make.UNKNOWN ../Make.linux
cd ../
vi Make.linux
```

上面最后一条命令用vi打开Make.linux文件，也可以用vscode打开。找到以下的行，将=后的内容修改为对应内容，#号后面的是注释。

```
ARCH          = linux      # 架构名字，与文件名Make.linux的后缀相同

MPdir         =             # mpi的安装路径，如果MPinc和MPlib没引用可以为空
MPinc         = -I/usr/include/x86_64-linux-gnu/mpich # MPI头文件路径
MPlib         = /usr/lib/x86_64-linux-gnu/libmpich.so  # MPI库文件路径

LAdir         =             # BLAS库的安装路径
LAinc         = -I/usr/include/x86_64-linux-gnu/     # BLAS库的头文件
LAlib         = /usr/lib/x86_64-linux-gnu/libopenblas.so # BLAS库的库文件
```

vi Make.top打开Make.top文件，同样找到ARCH的定义改为linux。

```
ARCH          = linux
```

make arch=linux 编译，注意检查输出是否有error，如有error则是编译失败，先检查以上配置是否有漏修改的，再根据提示信息查找原因，修改无误后先运行make clean_arch_all 清除已有的编译文件，再make arch=linux 编译。

编译成功会在~/hpl目录下生成bin/linux/xhpl可执行文件。

3.2 HPL运行

配置运行参数文件HPL.dat，相同的环境，参数不同，跑出来的结果不同，HPL的优化通常就是对HPL.dat参数进行优化，参数的设置跟系统的硬件配置有关。

```
cd ~/hpl/bin/linux
vi HPL.dat
```

根据需求修改，比如只用单核cpu的跑，设置P=1 and Q=1，然后运行./xhpl启动测试。运行结束后会将结果直接输出到屏幕。

如果要启动多核多进程测试，需要用mpirun运行。比如设置P=2 and Q=2，则需要4个进程。用mpirun启动的时候-n指定的进程数要不小于P*Q的值。

```
mpirun -n 4 ./xhpl
```

运行结果的开始和结束的输出如下。运行的tests的数量跟配置有关，是N、NB、PQ取值可能性的乘积。取最大的值作为最终的测量值。

```
$ linux »mpirun -n 4 ./xhpl
```

```
=====
HPLinpack 2.3  --  High-Performance Linpack benchmark  --  December 2, 2018
Written by A. Petitet and R. Clint Whaley,  Innovative Computing Laboratory, UTK
Modified by Piotr Luszczek, Innovative Computing Laboratory, UTK
Modified by Julien Langou, University of Colorado Denver
=====
```

An explanation of the input/output parameters follows:

```
T/V      : Wall time / encoded variant.
N        : The order of the coefficient matrix A.
NB       : The partitioning blocking factor.
P        : The number of process rows.
Q        : The number of process columns.
Time     : Time in seconds to solve the linear system.
Gflops   : Rate of execution for solving the linear system.
```

The following parameter values will be used:

```
N      :      29
NB     :      1
PMAP   : Row-major process mapping
P      :      2      1
Q      :      2      1
PFACT  : Left      Crout      Right
NBMIN  :      2      4
NDIV   :      2
RFACT  : Left      Crout      Right
BCAST  : 1ring
DEPTH  :      0
```

```
=====
T/V              N    NB    P    Q              Time              Gflops
-----
WR00R2R2        29     1     1     1              0.00              5.4842e-01
HPL_pdgesv() start time Tue Sep  3 17:44:49 2024

HPL_pdgesv() end time   Tue Sep  3 17:44:49 2024

=====
||Ax-b||_oo/(eps*(||A||_oo*||x||_oo+||b||_oo)*N)=  2.07824427e-02 ..... PASSED
=====
T/V              N    NB    P    Q              Time              Gflops
-----
WR00R2R4        29     1     1     1              0.00              5.4842e-01
HPL_pdgesv() start time Tue Sep  3 17:44:49 2024

HPL_pdgesv() end time   Tue Sep  3 17:44:49 2024

=====
||Ax-b||_oo/(eps*(||A||_oo*||x||_oo+||b||_oo)*N)=  2.07824427e-02 ..... PASSED
=====

Finished      36 tests with the following results:
              36 tests completed and passed residual checks,
               0 tests completed and failed residual checks,
               0 tests skipped because of illegal input values.

-----

End of Tests.
=====
```

主要参考：

- 1. [HPL Software Download and Installation](#)
- 2. [How to compile HPL LINPACK on Ubuntu 22.04](#)

4. 实验报告及要求

实验报告包括以下内容

- 1. 硬件配置：cpu型号、核数、频率，avx指令集版本，内存大小
- 2. CPU理论峰值计算
- 3. 软件环境：操作系统版本，MPI、数学库等软件的版本
- 4. 参数调优过程和说明，最优的测试结果，与CPU峰值性能的比例
- 5. 碰到的问题及解决方法