Benchmark

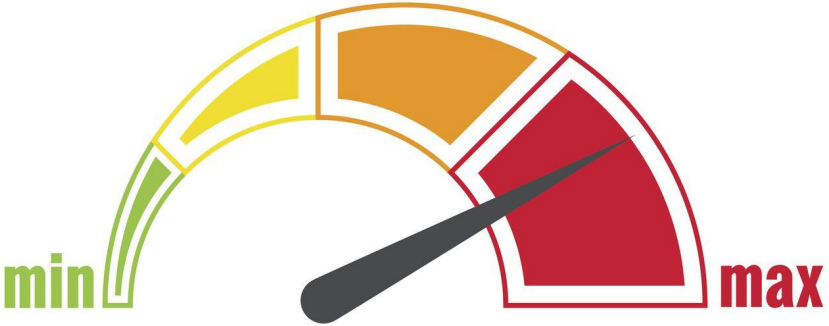
Outline

01 Introduce of benchmark

02 HPL

03 HPCG

04 Homework



Introduce of benchmark

Introduce of benchmark

Benchmark 是一種評估和比較產品、服務或過程性能的方法，從而識別出性能差距、發現改進的機會，並實施具體的改進措施。

對於電腦上的 Benchmark，是一種衡量和評估電腦硬體性能的方法，比較不同硬體或系統的Benchmark結果，使用者、開發者和製造商可以評估硬體產品的性能表現，並做出選擇或優化決策。

HPL

HPL

HPL(High Performance Linpack)是 Linpack 的升級版，用於測試超級電腦的運算效能。

HPL主要利用高斯消去法解決一個大規模的密集型線性代數方程組來測試系統的最大計算速度，並以浮點運算每秒次(flops，尤其是Gflops或Tflops)作為性能指標。

然而，HPL 只能反映系統的峰值性能，並不能全面描繪出所有類型計算工作負載下的性能表現。

HPL

openmpi

```
$ spack install openmpi@5.0.2%gcc
$ spack load openmpi@5.0.2
```

cmake

```
$ spack install cmake@3.27.9
$ spack load cmake@3.27.9
```

HPL

openblas (openmpi)

```
$ cd $HOME
$ spack load cmake@3.27.9
$ spack load openmpi@5.0.2
$ wget
  https://github.com/OpenMathLib/OpenBLAS/releases/download/v0.3.26/OpenBLAS
  -0.3.26.tar.gz
$ tar xf OpenBLAS-0.3.26.tar.gz
$ cd OpenBLAS-0.3.26 && rm -fr build
$ mkdir -p build && cd build
$ CC=mpicc CXX=mpicxx FC=mpifort cmake ..
  -DCMAKE_INSTALL_PREFIX=$HOME/OpenBLAS-0.3.26/build
$ make install -j $(nproc)
```

HPL

HPL install (在 work node 上 make)

```
$ cd $HOME
$ wget https://netlib.org/benchmark/hpl/hpl-2.3.tar.gz
$ tar xvf hpl-2.3.tar.gz && cd hpl-2.3/setup
$ sh make_generic
$ cp Make.UNKNOWN ../Make.linux && cd ..

    # Modify the arch mpdir and ladir in Make.linux

$ make arch=linux
$ cd bin/linux
```

HPL

modify Make.linux

```
$ ARCH         = linux
$ TOPdir       = $(HOME)/hpl-2.3
$
$ //OpenMPI路徑
$ MPdir        = <your path to OpenMPI>
$ MPinc        = -I$(MPdir)/include
$ MPlib        = $(MPdir)/lib/libmpi.so
$
$ //OpenBLAS路徑
$ LAdir        = $(HOME)/OpenBLAS-0.3.26/build
$ LAlib        = $(LAdir)/lib/libopenblas.a
```

HPL

HPL.dat 參數

```
$ 1             # of problems sizes (N)
$ 30            Ns
$
$ 1             # of NBs
$ 32            NBs
$
$ 1             # of process grids (P x Q)
$ 2             Ps
$ 2             Qs
```

這是要解的線性系統的大小。一般來 說，這個數值應該是硬體記憶體大小的 80 ~ 85%。
參考公式 N = sqrt(TotalMemory / 8)

這個數值表示計算中使用的區塊大小。一般來說，在 [32 .. 256] 這個區間可以得到不錯的效果。

這是一個表示計算機的處理器佈局的數字對。P 和 Q 應該接近相等，以使得處理器網格為方形。另外，P x Q 應該等於使用的 CPU 總數。

HPL

HPL.dat 參數

```
$ 1             # of panel fact
$ 2             PFACTs (0=left, 1=Crout, 2=Right)
$
$ 1             # of recursive stopping criterium
$ 4             NBMINs (>= 1)
$
$ 1             # of recursive panel fact.
$ 1             RFACTs (0=left, 1=Crout, 2=Right)
```

這些參數控制如何分發和管理數據以及如何進行因數分解。

HPL

run

```
$ mpirun -np 4 ./xhpl > output.txt
```

result

```
================================================================================
T/V                N    NB     P     Q               Time                 Gflops
--------------------------------------------------------------------------------
WR00C2R4          30    32     2     2               0.00             9.6447e-02
HPL_pdgesv() start time Tue Apr  2 13:05:26 2024

HPL_pdgesv() end time   Tue Apr  2 13:05:26 2024

--------------------------------------------------------------------------------
||Ax-b||_oo/(eps*(||A||_oo*||x||_oo+||b||_oo)*N)=   1.60609763e-02 ..... PASSED
================================================================================

Finished      1 tests with the following results:
              1 tests completed and passed residual checks,
              0 tests completed and failed residual checks,
              0 tests skipped because of illegal input values.
--------------------------------------------------------------------------------

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

在 output.txt 中可以看到最後的 Gflops, 並且要顯示 PASSED

HPCG

HPCG

HPCG（High Performance Conjugate Gradients）是一個用於衡量高性能計算系統（HPC）性能的基準測試套件，它旨在補充HPL的基準測試。

HPCG主要使用共軛梯度法（Conjugate Gradient, CG）來解決一個稀疏線性系統運算。

HPCG設計用來更全面地模擬現代科學計算工作負載，特別是那些涉及到稀疏矩陣運算的應用。

HPCG

HPCG install

```
$ cd $HOME
$ <load your mpi>
$ git clone https://github.com/hpcg-benchmark/hpcg.git
$ cd hpcg
$ mkdir build && cd build
$ ../configure Linux_MPI
$ make
$ cd bin
```

HPCG

hpcg.dat 參數

```
$ HPCG benchmark input file
$ Sandia National Laboratories; University of Tennessee, Knoxville
$ 48 48 48
$ 10
```

這是指定測試運行的總時間(以秒為單位)。

這些參數應於標量線性代數系統的尺寸，在 HPCG 中，這是由三個參數(nx、ny 和 nz)指定的，它們分別對應於空間三個維度的大小。
三個參數用空格隔開，也可以只寫一個，數字為 8 的倍數，最小不能低於 24。

HPCG

run

```
$ mpirun -np 4 ./xhpcg
```

result

```
GFLOP/s Summary=
GFLOP/s Summary::Raw DDOT=1.77862
GFLOP/s Summary::Raw WAXPBY=8.93784
GFLOP/s Summary::Raw SpMV=7.46318
GFLOP/s Summary::Raw MG=6.10651
GFLOP/s Summary::Raw Total=6.0551
GFLOP/s Summary::Total with convergence overhead=6.0551
GFLOP/s Summary::Total with convergence and optimization phase overhead=5.92518
User Optimization Overheads=
User Optimization Overheads::Optimization phase time (sec)=4.21e-07
User Optimization Overheads::Optimization phase time vs reference SpMV+MG time=1.70035e-05
DDOT Timing Variations=
DDOT Timing Variations::Min DDOT MPI_Allreduce time=0.1446
DDOT Timing Variations::Max DDOT MPI_Allreduce time=0.51161
DDOT Timing Variations::Avg DDOT MPI_Allreduce time=0.324198
Final Summary=
Final Summary::HPCG result is VALID with a GFLOP/s rating of=5.92518
Final Summary::HPCG 2.4 rating for historical reasons is=6.0551
Final Summary::Reference version of ComputeDotProduct used=Performance results are most likely suboptimal
Final Summary::Reference version of ComputeSPMV used=Performance results are most likely suboptimal
Final Summary::Reference version of ComputeMG used=Performance results are most likely suboptimal
Final Summary::Reference version of ComputeWAXPBY used=Performance results are most likely suboptimal
Final Summary::Results are valid but execution time (sec) is=10.6385
Final Summary::Official results execution time (sec) must be at least=1800
```

顯示最後的 final GF

要看到 HPCG result is VALID

Homework

依照上課簡報實作HPL及HPCG

HPL 參數要與簡報上的 N、NB、P、Q 不同並簡單解釋調整原因

HPCG 參數也要與簡報上的參數不同並簡單解釋調整原因

記得附上參數和result的截圖(參考簡報裡的部分)

deadline : 4 / 23