HPL

Jiarong song

December 2023

Contents

1 Software and hardware environment

1.1 CPU

Here are the information of my computer.

- CPU Model: AMD Ryzen 5 5600U with Radeon Graphics
- Number of Cores: 6
- Number of Threads: 12
- Base Frequency: 2.30 GHz
- L1 Cache: 384 KB
- L2 Cache: 3.0MB
- L3 Cache: 16.0MB

1.2 GPU

- GPU Model: AMD Radeon™ Graphics
- Driver Version: 30.0.13032.4000
- Driver Date: 2022/8/18
- DirectX Version: 12 (FL 12.1)
- Physical Location: PCI bus 5, device 0, function 0
- Dedicated GPU Memory: 0.5/2.0 GB
- Shared GPU Memory: 0.2/6.9 GB
- Total GPU Memory: 0.7/8.9 GB

1.3 Memory

- Total memory: 16.0 GB
- Speed: 3200 MHz
- Slots used: 2 out of 2
- Form factor: SODIMM
- Memory reserved for hardware: 2.1 GB
- Available: 7.2 GB
- Cached: 7.1 GB
- Committed: 10.4 out of 16.0 GB
- Paged pool: 565 MB
- Non-paged pool: 567 MB
- In use (Compressed): 6.7 GB (706 MB)

1.4 Linux

My computer operating system is windows, so I downloaded Linux using the WSL provided by windows.

- Distributor ID: Ubuntu
- Description: Ubuntu 22.04.2 LTS
- Release: 22.04
- Codename: jammy
- kernel version: 5.10.16.3-microsoft-standard-WSL2

1.5 HPL configuration

1.5.1 Compiler

- gcc version 11.4.0 (Ubuntu 11.4.0-1ubuntu1 22.04)
- vim version 8.2.4919

1.5.2 Math library

- BLAS-3.8.0
- CBLAS

1.5.3 HPL

- HPL-2.3

My HPL installation refers to the following link:

1. Configure the compiler:
   https://www.cnblogs.com/zhyantao/p/10614244.html
2. Install vector and matrix operations library functions:
   https://www.cnblogs.com/zhyantao/p/10614254.html
3. Linpack's HPL test (HPL Benchmark):
   https://www.cnblogs.com/zhyantao/p/10614238.html

1.6 HPL (CPU, CPU+GPU) compilation process

1.6.1 Select parameters

In the HPL.dat file, N represents the number and size of the matrix to be solved. The larger the matrix size N, the larger the proportion of effective calculations, and the higher the floating point processing performance of the system. However, the larger the matrix size will lead to more memory consumption. If the actual memory space of the system is insufficient, the use of cache and performance will be significantly reduced. It is optimal for the matrix to occupy about 80% of the total system memory, that is, $N \times N \times 8 =$ total system memory $\times$ 80% (the unit of total memory is bytes).
NB represents the size of the matrix chunks used during matrix solving. Block size has a great impact on performance, and the choice of NB is closely related to many factors in software and hardware. The selection of NB value is mainly based on actual testing to obtain the optimal value, which generally follows the following rules:

- NB cannot be too big or too small, generally less than 384.
- NB$\times$8 must be a multiple of cache lines.

- The size of NB is related to the communication method, matrix size, network, processor speed, etc.

P represents the number of processors in the horizontal direction, and Q represents the number of processors in the vertical direction. P×Q represents a two-dimensional processor grid. P×Q=number of system CPUs=number of processes. Under normal circumstances, one process corresponds to one CPU to obtain the best performance. For Intel® Xeon®, turning off Hyper-Threading can improve HPL performance. The values of P and Q generally follow the following rules:

- P $\leq$ Q. Generally, the value of P is less than Q, because the column communication volume (number of communications and communication data volume) is much larger than horizontal communication.
- P It is recommended to choose a power of 2. Horizontal communication in HPL uses binary exchange. When the number of horizontal processors P is a power of 2, the performance is optimal.

According this,the memory of my computer is 16GB,So the value of my parameter N is:

$$N_{textmax} = 0.8 \times \sqrt{\frac{16 \times 1024 \times 1024 \times 1024}{8}} \approx 37072$$

According to the physics core of my computer is 6,I first select P and Q as follows:

$$P = 2, Q = 3 \qquad P = 1, Q = 6 \qquad P = 6, Q = 1$$

Later, the best CPU performance was obtained based on the number of processes being 1, so I set P and Q as follows again for testing:

$$P = 1, Q = 1 \qquad P = 1, Q = 1 \qquad P = 1, Q = 1$$

The numerical setting of NB is as follows:

$$64 \qquad 128 \qquad 192 \qquad 256$$

The final parameters are as follows. Due to the influence of computer performance and time, this test cannot be completely completed.

```
The following parameter values will be used:

N : 30000 30000 30000 30000
NB : 64 128 192 256
PMAP : Row-major process mapping
```

```
P : 2 1 6
Q : 3 6 1
PFACT : Left Crout Right
NBMIN : 2 4
NDIV : 2
RFACT : Left Crout Right
BCAST : 1ring
DEPTH : 0
SWAP : Mix (threshold = 64)
L1 : transposed form
U : transposed form
EQUIL : yes
ALIGN : 8 double precision words
----------------------------------------------------------------
```

To save time, choose the value of the power of 2 closest to the value calculated above as the N value.Modify it as follows for subsequent testing:

```
----------------------------------------------------------------
The following parameter values will be used:

N : 32768
NB : 64 128 192 256
PMAP : Row-major process mapping
P : 1 1 1
Q : 1 1 1
PFACT : Left Crout Right
NBMIN : 2 4
NDIV : 2
RFACT : Left Crout Right
BCAST : 1ring
DEPTH : 0
SWAP : Mix (threshold = 64)
L1 : transposed form
U : transposed form
EQUIL : yes
ALIGN : 8 double precision words
----------------------------------------------------------------
```

Because the above-mentioned scale took too long to run, I later modified the parameters as follows based on network data:

```
----------------------------------------------------------------
The following parameter values will be used:

N : 32768
NB : 64 128 192 256
PMAP : Row-major process mapping
P : 1 1 1
```

```
Q : 1 1 1
PFACT : Left Crout Right
NBMIN : 2 4
NDIV : 2
RFACT : Left Crout Right
BCAST : 1ring
DEPTH : 0
SWAP : Mix (threshold = 64)
L1 : transposed form
U : transposed form
EQUIL : yes
ALIGN : 8 double precision words
```

Since I did not conduct CPU+GPU testing, I did not calculate the increase.

1.7 optimization

Since the scale of the problem solved is relatively small, I combined the network data to adjust the number of processes and NB at the same time to obtain the best performance. I did not get the optimal problem size of my computer, but combined with the GROMCS test, I personally judge that the best performance of my computer is when its process has its physical core number.

1.7.1 References during testing

- $https : //blog.csdn.net/weixin_4 2121713/article/details/100125356$
- $https : //blog.csdn.net/gyx1549624673/article/details/86551466$
- $https : //blog.csdn.net/yu_2 23/article/details/115983923$