

22级第一次总结

Groundhog

2023.04.08

ONEAPI的安装

1. 下载方法

鉴于国内尤其是百度网盘的带宽限制，最快的下载方法是从群文件中下载后利用局域网传输到Linux中。

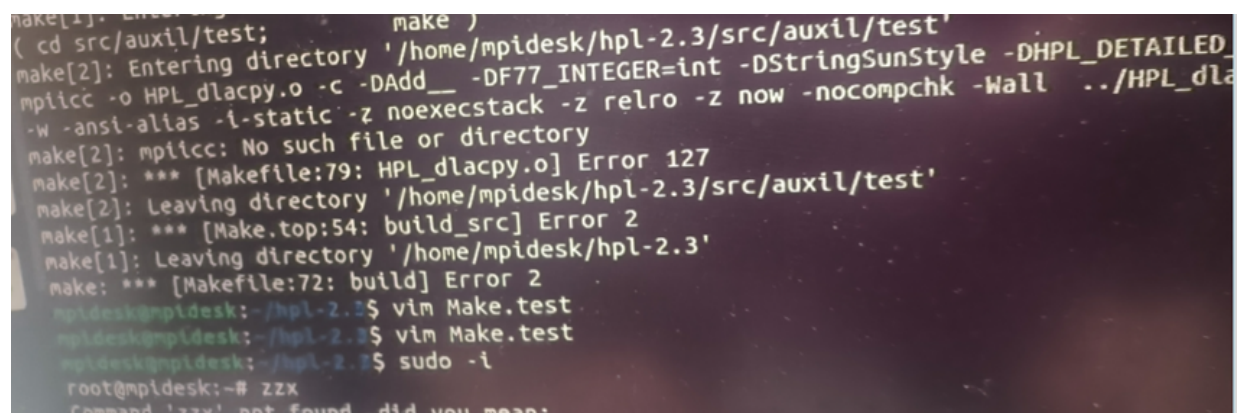
2. 安装过程复现：

1. 首先下载保存好两个sh文件
2. 对于新的虚拟机，我的建议是先换源。
3. 进行update与upgrade，进：行ssh及vim的安装
4. 安装依赖，依赖命令 apt -y install gcc cmake pkg-config build-essential。
5. 开始安装
6. 写入环境变量

3. 注意事项及部分报错解决方法：

(当然我个人不会遇到这么多问题的，只能说这些是多人汇总的)

1. mpiicc: no such file or directory等

A terminal window showing the installation of MPI. The user is in the directory /home/mpidesk/hpl-2.3/src/auxil/test. They run 'make' and then 'mpiicc -o HPL_dlapcy.o -c -DAdd__ -DF77_INTEGER=int -DStringSunStyle -DHPL_DETAILED_...'. The output shows several errors: 'mpiicc: No such file or directory', 'make[2]: *** [Makefile:79: HPL_dlapcy.o] Error 127', 'make[2]: Leaving directory '/home/mpidesk/hpl-2.3/src/auxil/test'', 'make[1]: *** [Makefile:54: build_src] Error 2', and 'make[1]: Leaving directory '/home/mpidesk/hpl-2.3''. The user then runs 'vim Make.test', 'vim Make.test', and 'sudo -i'. The prompt changes to 'root@mpidesk:~#', and the user enters 'zzx'. The terminal shows 'Command 'zzx' not found, did you mean:'.

出现如上原因是因为环境变量没有添加好，解决办法是安装时直接使用超级用户进行安装，避免出现普通用户的环境变量与超级用户不同的情况。

2. can't find -lstdc++ 链接库的问题，需要建立链接

ln -s /usr/lib/x86_64-linux-gnu/libstdc++.so.6 /usr/lib/x86_64-linux-gnu/libstdc++.so

3. _malloc__报错

```
/usr/include/stdio.h(315): error: attribute "__malloc__" does not take arguments  
__attribute_malloc__ __attr_dealloc_fclose __wur;
```

在gcc为11时，的确会遇到此类错误，降到9即可

4. 给忘了，好像是提示color的问题，解决办法：面向CSDN编程

HPL的安装

1. 安装复现

1. wget下载并解压
2. 复制备份make文件
3. 修改参数
 - a. arch=<name>
 - b. 更改实际hpl的位置
 - c. 改为-qopenapi
4. make arch=<name>
5. 测试xhpl

HPL的测试与调优

1. hpl原理及参数意义

HPL 采用分块 LU 算法,将每个分块分解成NB*NB的矩阵进行计算,以 Block-Cycle 方式分配到二维进程网格,LU 分解完成后,HPL 使用回代求解 x,并验证解的正确性。

至于参数意义：

Ns：问题矩阵大小规模

NBs：分块矩阵大小

PMAP process mapping

P,Q：水平与垂直方向处理器个数

NBIVS：一步分解产生的子分块个数

RFACTS：用来选择产生子数据块的递归分解方式

NBMINs：分解算法采用递归的块分解方法,当分解到的方块的列数等于其值时时,分解算法不再进行块分解而是直接进行向量矩阵运算

BCASTs：广播的算法

DEATHS：提供给算法 设置如何对当前块的后续快的更新方式

2. 环境参数：

Intel i7-12700H, 主频2.3GHz, 核数14, 轻薄本, 散热性能极差。

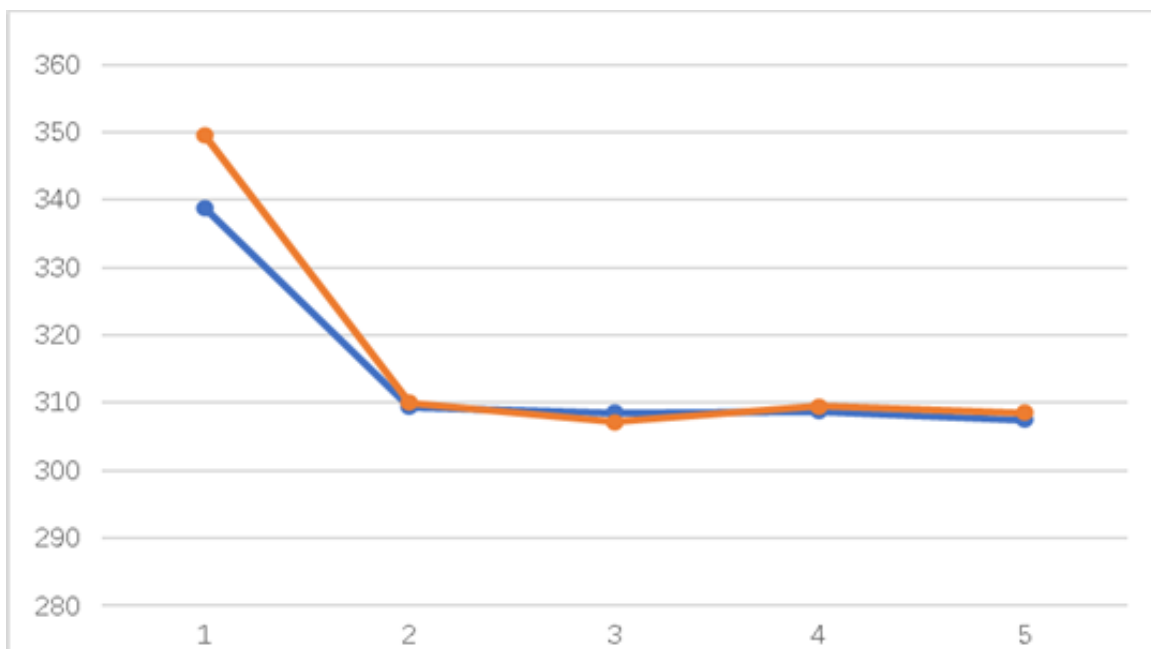
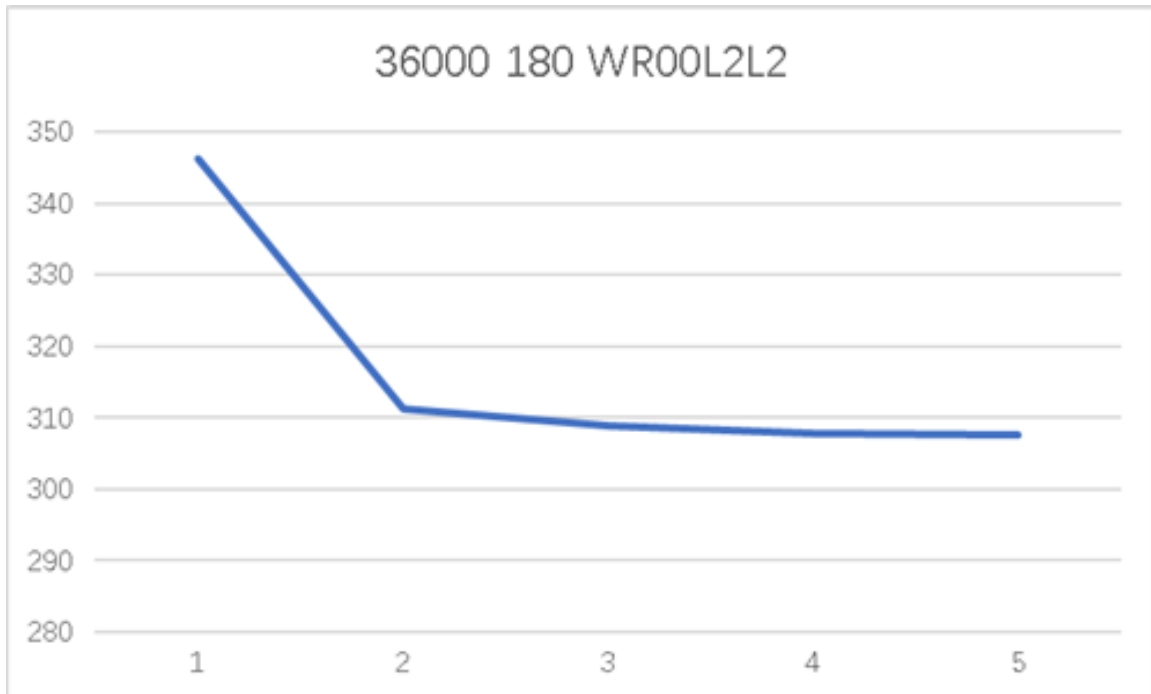
由于超线程技术、VM的虚拟机技术以及英特尔的大小核调度,我刚开始用VM跑出来的数据不太准确,我查看实际峰值比去理论峰值超过了百分之百的数据是,我认为其理论峰值不准确,遂转WSL。

关于理论峰值的计算,本机CPU每秒钟浮点运算次数为16。因此理论峰值为 $2.3 \times 14 \times 16 = 515.2$,此数值与英特尔官方给出的Gflops相同,得到验证。

3. 计算步骤

1. 计算方法：

我发现在个人PC上进行计算,由于轻薄本的原因,散热风口较小,物理降温效果不明显,其测试会因温度问题而导致降频,因此峰值会有约10%的削减,图二为N是42000及36000的数据,可见后期发热的数据在较小范围调参中基本不可用。



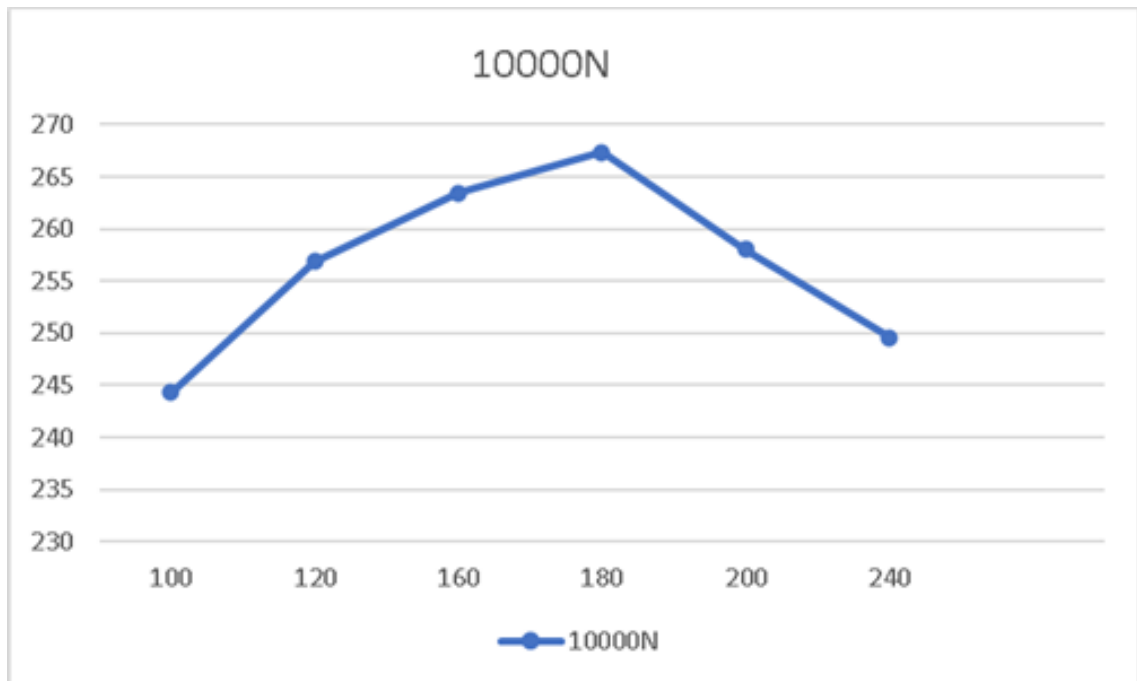
通过对CPU实时监控可知，其温度在后期稳定在80度左右，伴有CPU的降频。

因此我采用多次间断测试的方法，在一组测试后进行空闲处理，待冷却后再进行实验。

实验时关闭WiFi、火绒，降低屏幕亮度，后台CPU占比最大的除虚拟机外无其他可关闭运用。

2. 计算过程

a. 确定NBs的值，先开1000N，P1Q14，如下图可见为180为峰值。

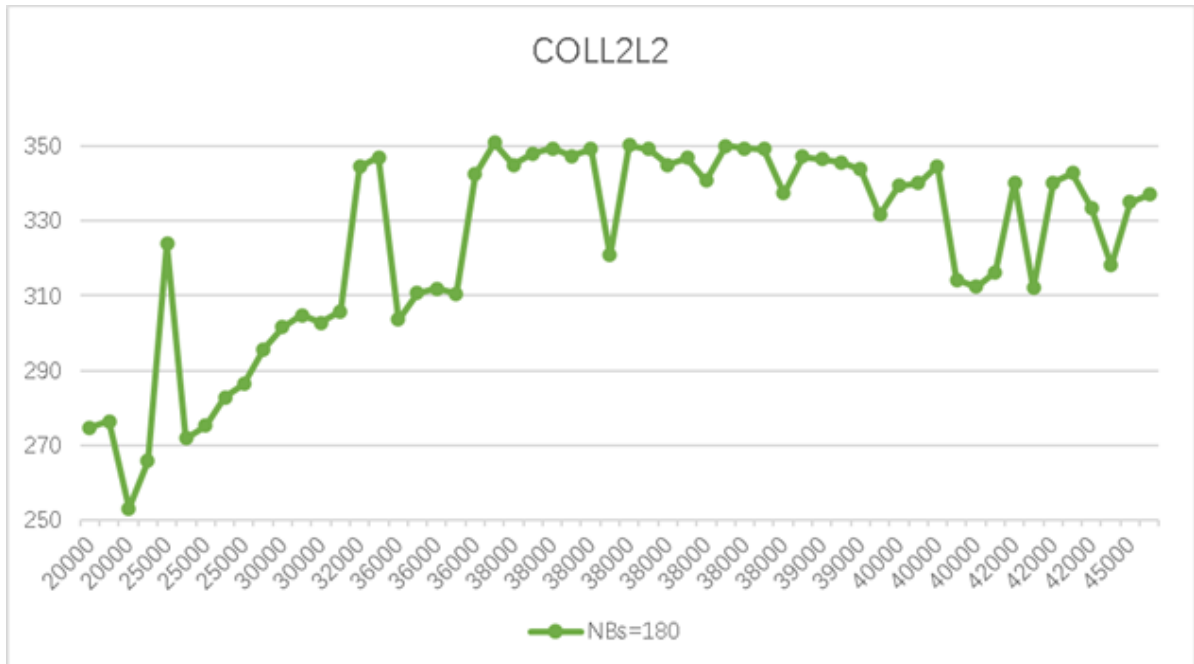


b. 首先进行Ns的确定，根据查阅文献可知，理想的Ns应满足

$$N * N * 8 = \text{内存大小} * a$$

其中 a 的取值介于80% ~ 90% 之间

而经我的实测，其达到峰值时占用内存约60%，原因可能是发热问题等。



由于发热等不确定因素，故重复了几次，45000~48000数据不比36000~40000好，由于作者excel水平有限，略去。

可以发现，在P1Q14，CL2L2时38000的表现最为稳定且优异，故确定。

c. 进行PQ的调试

确定好180-38000后，进行两组PQ的测试，P1Q14与P2Q7差距不大，温度波动影响>PQ影响。

d. 进行PMAP process mapping测试

结果：col>row （差约1%。。。）

e. 进行其他参数的微调

发现差别不大或原有参数最优。

f. 在WCL0L0 N38000 NBs180范围内进行微调，证实其值是最优的。

3. 计算结果

取最优的结果，比值68.10%

```

=====
T/V          N    NB    P    Q          Time          Gflops
-----
WC00L2L2     36000   180    1    14          88.65          3.5087e+02
HPL_pdgesv() start time Thu Apr  6 17:58:08 2023

HPL_pdgesv() end time   Thu Apr  6 17:59:37 2023

--VVV--VVV--VVV--VVV--VVV--VVV--VVV--VVV--VVV--VVV--VVV--VVV--VVV--VVV--
Max aggregated wall time rfact . . . :          0.47
+ Max aggregated wall time pfact . . . :          0.07
+ Max aggregated wall time mxswp . . . :          0.01
Max aggregated wall time update . . . :         76.97
+ Max aggregated wall time laswp . . . :          0.64
Max aggregated wall time up tr sv . . :          0.10
-----
||Ax-b||_oo/(eps*(||A||_oo*||x||_oo+||b||_oo)*N)=  1.28429426e-03 ..... PASSED
=====

```

相关参数


```
N      : 36000
NB     : 180
PMAP   : Column-major process mapping
P      : 1
Q      : 14
PFACT  : Left
NBMIN  : 2
NDIV   : 2
RFACT  : Left
BCAST  : 1ring
DEPTH  : 0
SWAP    : Mix (threshold = 64)
L1     : transposed form
U      : transposed form
EQUIL  : yes
ALIGN  : 8 double precision words
```

原始结果：

1	Column1	Column2	Column3	Column4	Column5	Column6	Column7	Column8
2	WR00C2L4	35	4	1	4	0	0.26393	0.179300%
3	WR00C2C4	35	4	1	4	0	0.26084	0.177201%
4	WR00L2C4	35	4	1	4	0	0.25993	0.176583%
5	WR00C2C2	34	2	1	4	0	0.12955	0.176019%
6	WR00C2R4	34	2	1	4	0	0.12908	0.175380%
7	WR00R2R2	34	2	1	4	0	0.12812	0.174076%
8	WR00R2C2	34	2	1	4	0	0.12776	0.173587%
9	WR00L2R2	35	4	1	4	0	0.25518	0.173356%
10	WR00R2L4	34	2	1	4	0	0.12691	0.172432%
11	WR00C2C4	34	2	1	4	0	0.12687	0.172378%
12	WR00C2R4	35	4	1	4	0	0.24718	0.167921%
13	WR00C2L2	35	4	1	4	0	0.24288	0.165000%
14	WR00L2C4	34	3	1	4	0	0.18215	0.164991%
15	WR00R2C4	34	2	1	4	0	0.12	0.163043%
16	WR00C2R2	34	2	1	4	0	0.11999	0.163030%
17	WR00L2R4	35	4	1	4	0	0.23817	0.161800%
18	WR00L2C2	34	3	1	4	0	0.17351	0.157165%
19	WR00R2L2	35	4	1	4	0	0.23021	0.156393%
20	WR00R2L2	34	2	1	4	0	0.11381	0.154633%

比值约0.18%

总结

由于个人PC的限制，其在数值上和调优方式上有所限制，服务器在N及PQ的选取上会与电脑上的不同。对于以后的学习要更基于理论化，了解缓存行、链接库以及CPU的组成等相关概念。

HPCG的安装

与hpl大致相同，略去。

参考文献

- [1] 罗水华,杨广文,张林波等.并行集群系统的Linpack性能测试分析[J].数值计算与计算机应用,2003(04):285-292.
- [2]张策,吴驰,龙涛.高性能计算集群Linpack测试与优化研究[J].信息与电脑(理论版),2021,33(18):14-17.

[3]黎雷生,杨文浩,马文静,张娅,赵慧,赵海涛,李会元,孙家昶.复杂异构计算系统HPL的优化[J].软件学报,2021,32(08):2307-2318.DOI:10.13328/j.cnki.jos.006003.

[4][HPL - A Portable Implementation of the High-Performance Linpack Benchmark for Distributed-Memory Computers \(netlib.org\)](#)

[5][HPL本机性能调优分析 - OZLIINEX's Blog \(ozline.icu\)](#)

[6][HPL - Wikipedia](#)

[7][GitHub - yunzhongOvO/Linpack-HPL: Linpack: configuration, install, optimization](#)

[8][Intel | Data Center Solutions, IoT, and PC Innovation](#)