lab3-Linux环境下调试与矩阵乘法优化

朱宏基　　220110131　　大二(上)

实验内容:

问题1:

要修改矩阵规模，需要更改矩阵的维度和循环的上限以允许处理更大的矩阵，发现 `test_MMult` 文件中的循环上限取决于头文件 `parameters.h` 的宏 `PLAST` ， 于是在该头文件中修改 `PLAST` 为1024即可，间距 `inc` 保持原来的值 `16` 不变。

运行结果如下:

```
hoshino_july@LAPTOP-S40B7Q22:~/hpc_practice/lab3-optimize-gemm/how-to-optimize-gemm$ make run
make all
make[1]: Entering directory '/home/hoshino_july/hpc_practice/lab3-optimize-gemm/how-to-optimize-gemm'
make clean;
make[2]: Entering directory '/home/hoshino_july/hpc_practice/lab3-optimize-gemm/how-to-optimize-gemm'
rm -f *.o *~ core *.x
make[2]: Leaving directory '/home/hoshino_july/hpc_practice/lab3-optimize-gemm/how-to-optimize-gemm'
make test_MMult.x
make[2]: Entering directory '/home/hoshino_july/hpc_practice/lab3-optimize-gemm/how-to-optimize-gemm'
gcc -Wall -g  -c test_MMult.c -o test_MMult.o
gcc -Wall -g  -c MMult0.c -o MMult0.o
gcc -Wall -g  -c copy_matrix.c -o copy_matrix.o
gcc -Wall -g  -c compare_matrices.c -o compare_matrices.o
gcc -Wall -g  -c random_matrix.c -o random_matrix.o
gcc -Wall -g  -c dclock.c -o dclock.o
gcc -Wall -g  -c REF_MMult.c -o REF_MMult.o
gcc -Wall -g  -c print_matrix.c -o print_matrix.o
gcc test_MMult.o MMult0.o copy_matrix.o compare_matrices.o random_matrix.o dclock.o REF_MMult.o print_matrix.o -lm \
        -o test_MMult.x
make[2]: Leaving directory '/home/hoshino_july/hpc_practice/lab3-optimize-gemm/how-to-optimize-gemm'
make[1]: Leaving directory '/home/hoshino_july/hpc_practice/lab3-optimize-gemm/how-to-optimize-gemm'
echo "version = 'MMult0';" > output_MMult0.m
echo $OMP_NUM_THREADS
1
./test_MMult.x >> output_MMult0.m
cp output_MMult0.m output_old.m
cp output_MMult0.m output_new.m
hoshino_july@LAPTOP-S40B7Q22:~/hpc_practice/lab3-optimize-gemm/how-to-optimize-gemm$ ls
MMult0.c    REF_MMult.c         copy_matrix.c  makefile         parameters.h      random_matrix.c  test_MMult.x
MMult0.o    REF_MMult.o         copy_matrix.o  output_MMult0.m  print_matrix.c    random_matrix.o  test_python_dgemm.py
MMult1.c    compare_matrices.c  dclock.c       output_new.m     print_matrix.o    test_MMult.c
PlotAll.py  compare_matrices.o  dclock.o       output_old.m     python_dgemm.jpg  test_MMult.o
hoshino_july@LAPTOP-S40B7Q22:~/hpc_practice/lab3-optimize-gemm/how-to-optimize-gemm$ ./test_MMult.x
MY_MMult = [
4 1.280000e-01 0.000000e+00
```

...

```
484 1.688642e-01 0.000000e+00
500 3.233793e-01 0.000000e+00
516 3.538732e-01 0.000000e+00
532 2.349667e-01 0.000000e+00
548 3.514893e-01 0.000000e+00
564 2.810218e-01 0.000000e+00
580 3.688841e-01 0.000000e+00
596 3.452219e-01 0.000000e+00
612 3.472755e-01 0.000000e+00
628 1.694843e-01 0.000000e+00
644 3.924507e-01 0.000000e+00
660 3.395380e-01 0.000000e+00
676 3.990997e-01 0.000000e+00
692 4.205404e-01 0.000000e+00
708 3.638099e-01 0.000000e+00
724 4.050078e-01 0.000000e+00
740 2.357598e-01 0.000000e+00
756 3.737772e-01 0.000000e+00
772 3.509555e-01 0.000000e+00
788 3.122557e-01 0.000000e+00
804 2.056657e-01 0.000000e+00
820 3.121753e-01 0.000000e+00
836 3.196549e-01 0.000000e+00
852 4.029098e-01 0.000000e+00
868 2.587372e-01 0.000000e+00
884 2.654602e-01 0.000000e+00
900 3.666626e-01 0.000000e+00
916 3.789244e-01 0.000000e+00
932 3.568820e-01 0.000000e+00
948 4.151536e-01 0.000000e+00
964 4.096675e-01 0.000000e+00
980 4.159811e-01 0.000000e+00
996 4.244709e-01 0.000000e+00
1012 4.240198e-01 0.000000e+00
];
```

问题2:

通过阅读gcc手册，对编译优化选项有了大致了解。编译优化选项用于告诉编译器如何优化生成的目标代码以提高程序的性能或减少代码的大小。
其中:

- `-O0` 表示关闭优化生成易于调试的目标代码。这样生成的代码会比较慢，但可以方便调试。
- `-O1` 表示基本优化， `-O2` 表示更多优化， `-O3` 表示最大优化级别。随着级别的增加，优化程度也会增加，但也可能增加编译时间。

通过修改 `makefile` 文件中的 `CFLAGS` ， 分别获得了在O0,O1,O2,O3不同的优化选项下 `MMult0.c` 实现的 `naive gemm` 的性能表现。
以下是4种不同优化选项下的性能表现:

- O0:

```
MY_MMult = [
4 1.280000e-01 0.000000e+00
20 5.517241e-01 0.000000e+00
36 5.621205e-01 0.000000e+00
52 5.624320e-01 0.000000e+00
68 4.624000e-01 0.000000e+00
84 4.762587e-01 0.000000e+00
100 3.348962e-01 0.000000e+00
116 4.358827e-01 0.000000e+00
132 3.136890e-01 0.000000e+00
148 3.432465e-01 0.000000e+00
164 3.753675e-01 0.000000e+00
180 3.765617e-01 0.000000e+00
196 3.970018e-01 0.000000e+00
212 3.800609e-01 0.000000e+00
228 4.159085e-01 0.000000e+00
244 3.696242e-01 0.000000e+00
];
```

- O1:

```
MY_MMult = [
4 inf 0.000000e+00
20 1.142857e+00 0.000000e+00
36 5.905823e-01 0.000000e+00
52 6.792657e-01 0.000000e+00
68 7.687824e-01 0.000000e+00
84 7.064410e-01 0.000000e+00
100 7.616146e-01 0.000000e+00
116 5.672891e-01 0.000000e+00
132 4.649218e-01 0.000000e+00
148 6.193718e-01 0.000000e+00
164 6.626521e-01 0.000000e+00
180 6.669716e-01 0.000000e+00
196 7.111386e-01 0.000000e+00
212 7.377282e-01 0.000000e+00
228 6.395096e-01 0.000000e+00
244 6.835812e-01 0.000000e+00
];
```

- O2:

```
MY_MMult = [
4 inf 0.000000e+00
20 2.666667e+00 0.000000e+00
36 2.521946e+00 0.000000e+00
52 2.286309e+00 0.000000e+00
68 1.776452e+00 0.000000e+00
84 1.820903e+00 0.000000e+00
100 1.557632e+00 0.000000e+00
116 1.751847e+00 0.000000e+00
132 1.886766e+00 0.000000e+00
148 1.931363e+00 0.000000e+00
164 1.682924e+00 0.000000e+00
180 1.813433e+00 0.000000e+00
196 1.544203e+00 0.000000e+00
212 1.738709e+00 0.000000e+00
228 2.146388e+00 0.000000e+00
244 1.911670e+00 0.000000e+00
];
```

- O3：

```
MY_MMult = [
4 inf 0.000000e+00
20 2.666667e+00 0.000000e+00
36 3.010065e+00 0.000000e+00
52 2.899134e+00 0.000000e+00
68 2.884697e+00 0.000000e+00
84 2.891239e+00 0.000000e+00
100 2.127660e+00 0.000000e+00
116 2.357849e+00 0.000000e+00
132 2.395800e+00 0.000000e+00
148 2.133460e+00 0.000000e+00
164 1.960420e+00 0.000000e+00
180 1.607719e+00 0.000000e+00
196 1.895654e+00 0.000000e+00
212 2.006344e+00 0.000000e+00
228 1.752399e+00 0.000000e+00
244 2.092141e+00 0.000000e+00
];
```

问题3:

通过在 `test_MMult.c` 文件中直接设计了 `OpenBLAS_MMult` 函数，并在 `makefile` 文件中的编译处加上 `-lopenblas` 以链接其 `OpenBLAS` 库实现将 `openblas` 的实现集成到框架代码，并在 `test_MMult.c` 文件中增加对调用 `OpenBLAS_MMult` 函数所得的性能分析进行输出，得到运行结果如下：

```
MY_MMult = [
4 1.280000e-01 0.000000e+00
20 2.539683e-01 0.000000e+00
36 2.542561e-01 0.000000e+00
52 2.434771e-01 0.000000e+00
68 1.890183e-01 0.000000e+00
84 9.226401e-02 0.000000e+00
100 1.263504e-01 0.000000e+00
116 2.297970e-01 0.000000e+00
132 3.003549e-01 0.000000e+00
148 3.492370e-01 0.000000e+00
164 3.097464e-01 0.000000e+00
180 3.734631e-01 0.000000e+00
196 3.805006e-01 0.000000e+00
212 4.244060e-01 0.000000e+00
228 3.658698e-01 0.000000e+00
244 4.099038e-01 0.000000e+00
];
OpenBLAS_MMult = [
4 2.327273e-03 2.220446e-16
20 4.000000e+00 1.776357e-15
36 6.665143e+00 3.552714e-15
52 1.406080e+01 3.552714e-15
68 1.591658e-01 7.105427e-15
84 9.445482e-01 8.881784e-15
100 4.106776e+00 9.769963e-15
116 8.878817e-01 8.881784e-15
132 8.519978e-01 1.065814e-14
148 7.592019e+00 1.421085e-14
164 1.673983e+01 1.421085e-14
180 3.719388e+00 1.776357e-14
196 1.479135e+00 1.243450e-14
212 2.936249e+01 1.776357e-14
228 3.058671e+01 1.953993e-14
244 5.070431e+01 1.421085e-14
];
```