

Lab4-GEMM 通用矩阵乘法

实验思路

MPI 进程并行

采用了4节点16进程的进程并行，首先，由0号进程负责矩阵A，B的转置和相加操作，由1号进程负责矩阵C2的清零，然后将矩阵通过MPI_Bcast的方式传给其他进程，各自进程负责各自的计算任务；当全部进程计算结束后，再通过MPI_Gather传给0号进程，汇总相加算出最后结果，另外2号进程还要处理部分的计算任务，计算完成后再传给0号进程进行汇总

OMP 进程并行

指定了线程数为8，使用 `#pragma omp parallel for` 的方式加速for循环

Blocking

把大矩阵的乘法转成小矩阵的乘法，提高缓存利用率，块的大小设置为256，64的长方形矩阵

4*4的计算单元

每次计算时同时计算4*4的小单元，加速计算

编译选项

开启了-Ofast，让编译器向量化代码；使用了-fopenmp，开启openmp的线程并行；开启了-favx2，允许使用SIMD指令集

优化缓存

优化循环顺序，减小cache miss；baseline的代码中，刚开始的矩阵转置，也减小了cache miss的几率

运行参数

```
1. 节点4 进程16 线程8
make
I_MPI_PMI_LIBRARY=/usr/lib/x86_64-linux-gnu/libpmi.so.0 srun -N 4 -n 16 -c 8
./gemm
2. 节点4 进程16 线程8
make run
```

实验结果

实验次数	n=1	n=2	n=4	n=8
1	$8.4068 * 10^4$	$8.5598 * 10^4$	$8.5978 * 10^4$	$8.5689 * 10^4$
2	$8.6022 * 10^4$	---	---	---
3	$8.5518 * 10^4$	---	---	---
平均	$8.5202 * 10^4$	$8.5598 * 10^4$	$8.5978 * 10^4$	$8.5689 * 10^4$

单位均为Mops