

高性能计算程序设计基础 秋季 2021

提交格式说明

按照实验报告模板填写报告，需要提供源代码及代码描述至 <https://easyhpc.net/course/129>。实验报告模板使用 PDF 格式，命名方式为高性能计算程序设计_学号_姓名。如果有问题，请发邮件至 jiangjzh6@mail2.sysu.edu.cn, liuyh73@mail2.sysu.edu.cn 询问细节。

1. 通用矩阵乘法

数学上，一个 $m \times n$ 的矩阵是一个由 m 行 n 列元素排列成的矩形阵列。矩阵是高等代数中常见的数学工具，也常见于统计分析等应用数学学科中。矩阵运算是数值分析领域中的重要问题。

通用矩阵乘法（GEMM）通常定义为：

$$C = AB$$

$$C_{m,n} = \sum_{k=1}^N A_{m,k} B_{k,n}$$

请根据定义用 C/C++ 语言实现一个矩阵乘法：

题目：用 C/C++ 语言实现通用矩阵乘法

输入：M, N, K 三个整数（512 ~ 2048）

问题描述：随机生成 $M \times N$ 和 $N \times K$ 的两个矩阵 A, B, 对这两个矩阵做

乘法得到矩阵 C.

输出: A,B,C 三个矩阵以及矩阵计算的时间

2. 通用矩阵乘法优化

对上述的矩阵乘法进行优化, 优化方法可以分为以下两类:

- 1) 基于算法分析的方法对矩阵乘法进行优化, 典型的算法包括 Strassen 算法和 Coppersmith–Winograd 算法.
- 2) 基于软件优化的方法对矩阵乘法进行优化, 如循环拆分向量化和内存重排
- 3) 优化后的矩阵乘法与 Intel MKL 函数库的矩阵乘法函数, 进行性能对比 (相同规模矩阵乘法完成时间), 并试着解释原因。

Intel MKL 矩阵乘法参考资料:

<https://software.intel.com/content/www/us/en/develop/documentation/mkl-tutorial-c/top/multiplying-matrices-using-dgemm.html>

<https://stackoverflow.com/questions/23619713/matrix-multiplication-with-mkl>

在 Linux (Ubuntu) 上安装 MKL 的参考方法 (也可以在网络上搜索其他方法)

然后直接使用apt来安装mkl2020.2:

```
1 | sudo apt-get install intel-mkl-64bit-2020.2
2 |
```

最后source一下大功告成:

```
1 | source /opt/intel/compilers_and_libraries_2020/linux/mkl/bin/mklvars.sh intel64 ilp64
2 |
```

实验要求: 对优化方法进行详细描述, 并提供优化后的源代码, 以及与 GEMM 的计算时间对比

3. 进阶: 大规模矩阵计算优化

进阶问题描述: 如何让程序支持大规模矩阵乘法?

考虑两个优化方向

- 1) 性能, 提高大规模稀疏矩阵乘法性能;
- 2) 可靠性, 在内存有限的情况下, 如何保证大规模矩阵乘法计算完成 ($M, N, K \gg 100000$), 不触发内存溢出异常。

对优化方法及思想进行详细描述, 提供大规模矩阵计算优化代码可加分。

References:

[1]{GEMM 优化}

<https://jackwish.net/2019/gemm-optimization.html>

[2] {矩阵说明}

<https://zh.wikipedia.org/wiki/%E7%9F%A9%E9%98%B5>