



南開大學  
Nankai University

计算机学院  
并行程序设计第 2.1 次作业

矩阵与向量内积

姓名：丁屹  
学号：2013280  
专业：计算机科学与技术

2022 年 3 月 13 日

## 目录

<b>1 问题</b>	<b>2</b>
<b>2 程序实现</b>	<b>2</b>
<b>3 实验平台配置</b>	<b>2</b>
<b>4 实验方案设计</b>	<b>2</b>
4.1 测试脚本 . . . . .	2
4.2 测试数据 . . . . .	3
<b>5 参考文献</b>	<b>4</b>

## 1 问题

计算给定  $n \times n$  矩阵的每一列与给定向量的内积，考虑两种算法设计思路：

1. 逐列访问元素的平凡算法
2. cache 优化算法

## 2 程序实现

源码链接：<https://github.com/ArcanusNEO/Parallel-Programming/tree/master/1/0>

头文件位于 inc/，源文件位于 src/

### 逐列访问平凡算法

```
1 for (int i = 1; i <= matrix.col(); ++i)
2     for (int j = 1; j <= matrix.row(); ++j)
3         ans += (long long) matrix(j, i) * vec(i);
```

### cache 优化算法

```
1 for (int i = 1; i <= matrix.row(); ++i)
2     for (int j = 1; j <= matrix.col(); ++j)
3         ans += (long long) matrix(i, j) * vec(j);
```

- 为了便于调整数据规模的同时保证数据分布紧凑，矩阵采用一维数组模拟，封装到 `matrix_t` 类中，使用 `operator()` 访问元素
- 使用 C++11 的 `chrono::high_resolution_clock` 高精度计时函数测量运行时间
- `ordinary` 采用逐列访问的平凡算法
- `cache` 采用 cache 优化算法
- 使用 `cmake` 构建

## 3 实验平台配置

华为鲲鹏 arm 平台部分硬件参数如表 1 所示。

系统环境为 4.14.0 内核的 openEuler，本次实验使用基于 clang 的华为 bisheng 编译器构建。

## 4 实验方案设计

### 4.1 测试脚本

测试脚本位于 bin/，“run”是 x86 架构下的脚本，“run-pbs”是鲲鹏服务器平台的脚本

CPU Maximum Frequency	2600 MHz
CPU Minimum Frequency	200 MHz
L1d 缓存	64 KiB
L1i 缓存	64 KiB
L2 缓存	512 KiB
L3 缓存	49152 KiB
内存大小	191.3 GiB

表 1: 鲲鹏服务器硬件配置信息

## 4.2 测试数据

使用 gen-data 生成数据, 规模  $n$  作为第一个参数传入, 使用 mt1937 生成随机数。

生成了一组测试文件位于 res/, 文件名形如  $n.in$

使用 conf/in.conf 配置输入数据路径和重复测试次数, 其路径作为待测程序的第一个参数传入。

## 5 参考文献

[1][3][2][5][6][4]

## 参考文献

- [1] Alexis Zhang. Apple m1 wikipedia. [https://zh.wikipedia.org/wiki/Apple\\_M1](https://zh.wikipedia.org/wiki/Apple_M1), 2020.
- [2] Andrei Frumusanu. The 2020 mac mini unleashed: Putting apple silicon m1 to the test. <https://www.anandtech.com/print/16252/mac-mini-apple-m1-tested>, 2020.
- [3] Andrei Frumusanu. Apple announces the apple silicon m1: Ditching x86 - what to expect, based on a14. <https://www.anandtech.com/show/16226/apple-silicon-m1-a14-deep-dive>, 2020.
- [4] Erik Engheim. Why is apple's m1 chip so fast? <https://debugger.medium.com/why-is-apples-m1-chip-so-fast-3262b158cba2>, 2020.
- [5] Veedrac. Measures microarchitectural details. <https://github.com/Veedrac/microarchitecturometer>, 2020.
- [6] 木头龙. 如何看待苹果 m1 芯片跑分超过 i9? . <https://www.zhihu.com/question/429951450>, 2020.