



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 14 日

目录

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 arm 平台硬件配置信息

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 ?? 所示。arm 服务器系统环境为 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。

CPU 型号	AMD Ryzen 7 4800HS with Radeon Graphics
CPU Maximum Frequency	2900 MHz
CPU Minimum Frequency	1400 MHz
L1d 缓存	256 KiB
L1i 缓存	256 KiB
L2 缓存	4 MiB
L3 缓存	8 MiB
内存大小	16 GiB

表 2: 本地 x86 平台硬件配置信息

本地 x86 平台部分硬件参数如表 ?? 所示。本地 x86 系统环境为 5.16.14 内核的 Arch Linux，本次实验使用 GNU GCC 编译，并且使用 perf 进行性能测试。

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

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。

4.3 测试方法

分别测试 ordinary 和 cache 两个程序。读入定义测试输入文件路径和重复测试遍数的配置文件以便自动完成测试。程序会向标准输出打印测试结果。

重复测试代码

```

1  for (int _counter = 0; _counter < T; ++_counter) { // T 为重复测试遍数
2      ans      = 0;
3      auto t1 = chrono::high_resolution_clock::now();
4      func(ans, matrix, vec);
5      auto t2 = chrono::high_resolution_clock::now();
6      auto sec = chrono::duration_cast<chrono::duration<double>>(t2 - t1);
7      ret += sec.count() / T;
8  }
```