

# LAB1 test report

BY 小组：崩 撤 卖 溜

## 1. 实验概要

求解给出的数独求解问题，然后通过解决此问题来比较多线程和单线程之间的性能差异，和不同代码实验同一功能产生的性能差异，以及多线程在不同的硬件环境下的性能差异。

### 1.1 程序输入

在控制台接受用户输入，此输入应为某一目录下的数独解密文件，且包含若个数独问题，并且这些问题格式均固定。

### 1.2 程序输出

程序按照输入的数独顺序来输出对应的答案。

### 1.3Sudoku 算法

我们组从 4 个不同算法 BASIC,DANCE,MINA,MINAC 中选择了最快速的 DANCE 算法，其中最慢的是 BASIC。

### 1.4 性能指标

性能指标即时间开销，也就是根据将单个文件中所有数独问题均解决，并且按照正确的顺序（即输入的顺序）输出所花费的时间进行比较和测试。

一般使用加速比来进行比较并行和串行程序的性能差异。

### 1.5 实验环境

实验环境 1： LINUX 内核版本是 3.13.0-32-generic.

Intel(R) Core(TM) i5-7300HQ CPU @ 2.50GHz 2.50 GHz

| 设备            | 摘要    | 处理器              |
|---------------|-------|------------------|
| 内存            | 1 GB  | 处理器数量(P): 1      |
| 处理器           | 1     | 每个处理器的内核数量(C): 1 |
| 硬盘 (SCSI)     | 20 GB | 处理器内核总数: 1       |
| CD/DVD (SATA) | 自动检测  |                  |

实验环境 2: LINUX 内核版本是 5.0.0-38-generic。

| 处理器 Intel(R) Core(TM) i5-9300H CPU @ 2.40GHz 2.40 GHz |                          |                  |
|---|--------------------------|------------------|
| 设备  | 摘要                       | 处理器              |
| 内存  | 2 GB                     | 处理器数量(P): 2      |
| 处理器   | 2                        | 每个处理器的内核数量(C): 1 |
| 硬盘 (SCSI)   | 20 GB                    | 处理器内核总数: 2       |
| CD/DVD (SATA)   | 正在使用文件 ubuntu-19.04-d... |                  |

## 1.6 代码实现版本

CODE1:本实验提供的代码，单线程，注：给出的代码有些问题，我们加以改动了很多地方。

CODE2: 我们小组多人共同编写的代码，和 CODE1 有较大的不同，最终版本的代码大概几百行左右，未超过一千行。

默认使用 CODE2。

## 2. 性能测试

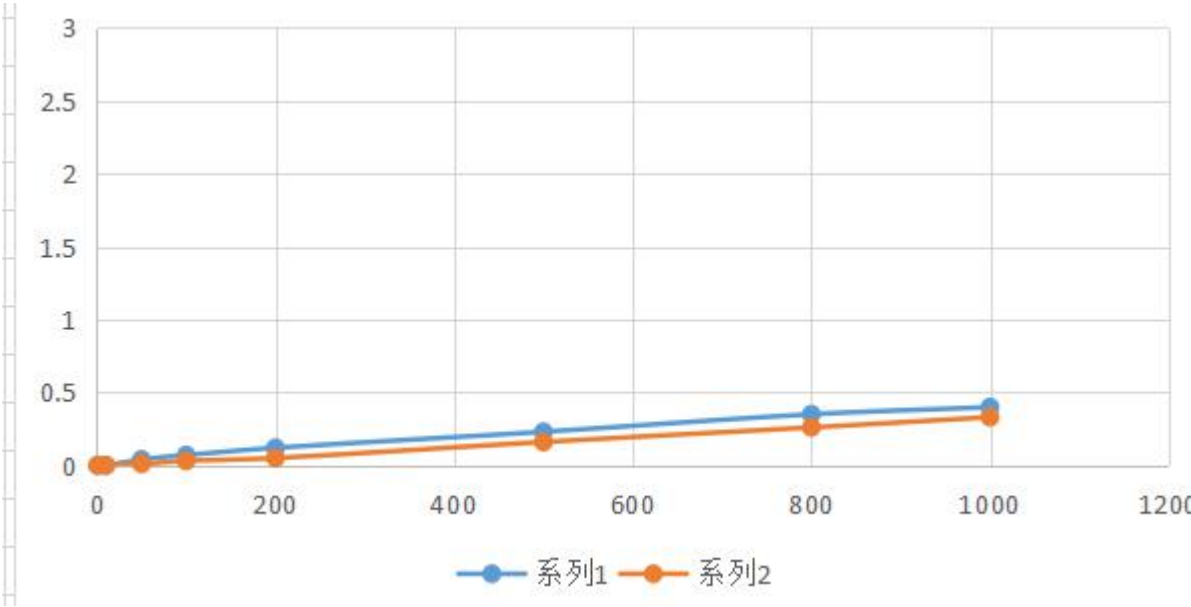
我们将从单线程和多线程不同代码的性能差异、以及不同环境中同一个程序的性能差异来测试和比较。

### 2.1 多线程与单线程不同代码性能比较

首先，CODE2 比 CODE1 多了很多代码段，随着数独问题数量变多，CODE2 和 CODE1 的差距在逐步缩小，后续会基本持平，在 1000 规模

问题的时候，可以发现，CODE2 创建数量多导致其耗时很久，并且输出结果之前要先保存，最后才按照顺序输出，而 CODE1 则是做完一个输出一个，如果数据规模继续增大，5000，10000 甚至更多，CODE1 的时间开销可能比 CODE2 的时间开销要小。

我们使用 8 组不同规模的数据进行测试比较，进而得出如下的曲线图。注：蓝色为 CODE1，黄色为 CODE2，横坐标为数据规模，纵坐标为时间开销，单位 S。



## 2.2 不同硬件环境性能测试

在实验环境 1 和实验环境 2 下进行 8 组数据测试，结果如下：

注：横坐标为数据规模，纵坐标为时间开销，蓝色为环境 2，黄色为环境 1。

可以看出，在小于 800 规模的数据，两者相差不大，但是环境 2 的时间开销更小一些，性能更好，但是超过 800 规模之后，会发现，环境 2 的时间开销会变大，并且超过环境 1，如果数据规模更大，甚至达

到 100000，那么两者的差距会更加明显，这时环境 1 的性能表现就要优于环境 2 了。

