



云计算性能分析报告

实 验 名 称: 实验一

组长 姓 名: 田志鹏-201808010404

学 生 姓 名: 沈祥振-201808010427

学 生 姓 名: 谭江浩-201808010405

学 生 姓 名: 罗凯洋-201808010527

完 成 时 间: 2021年4月8日

信息科学与工程学院

一、 实验概要

多线程编程是高性能编程的技术之一，实验 1 将针对数独求解问题比较多线程与单线程的性能差异、同一功能不同代码实现的性能差异以及多线程在不同硬件环境下的性能差异。

库网址：https://github.com/tzphh/CloudComputing_Labs.git

最终实验结果路径：CloudComputingLabs/Lab1/

1.1 程序输入

程序将在控制台接收用户输入，该输入应为某一目录下的一个数独谜题文件，该文件包含多个数独谜题，每个数独谜题按固定格式存储在该文件中。

接受一个输入文件名，并且输入文件大小小于 100MB。

1.2 程序输出

实验中把数独的解按与输入相对应的顺序写入到 `stdout` 文件中。

1.3 Sudoku 算法

实验共提供了 4 种不同的 Sudoku 求解算法：BASIC,DANCE,MINA 和 MINAC。

其中，DANCE 算法速度最快，BASIC 算法速度最慢。

实验中选用的是源代码默认的 BASIC 算法。

1.4 性能指标

实验以求解完单个输入文件里的所有数独题并把数独的解按顺序写入文件所需要的时间开销作为性能指标。

一般而言，可以用加速比直观地表示并行程序与串行程序之间的性能差异（加速比：串行执行时间与并行执行时间的比率，是串行与并行执行时间之间一个具体的比较指标）。

在用户输入文件路径进行运算后，在输出结果最后会获取运行时间，单位 ms。

1.5 实验环境

Linu 内核版本为 3.13.0-32-generic;

4GB 内存;

CPU 型号为 Intel(R) Core(TM) i5-7300HQ CPU @ 2.5GHz 2.50GHz,

共有 2 个核心 CPU;

不使用超线程技术

```

(base) tzp@tzp-virtual-machine:~$ lscpu
架构: x86_64
CPU 运行模式: 32-bit, 64-bit
字节序: Little Endian
Address sizes: 45 bits physical, 48 bits virtual
CPU: 2
在线 CPU 列表: 0,1
每个核的线程数: 1
每个座的核数: 1
座: 2
NUMA 节点: 1
厂商 ID: GenuineIntel
CPU 系列: 6
型号: 158
型号名称: Intel(R) Core(TM) i5-7300HQ CPU @ 2.50GHz
步进: 9
CPU MHz: 2496.004
BogoMIPS: 4992.00
超管理器厂商: VMware
虚拟化类型: 完全
L1d 缓存: 64 KiB
L1i 缓存: 64 KiB
L2 缓存: 512 KiB
L3 缓存: 12 MiB
NUMA 节点0 CPU: 0,1

```

1.6 代码实现版本

为适应多线程而在 Code1 上进行了一系列的修改和增添而成。

在 Code2 中，可通过参数的调节而控制线程数量（指定线程数量为 N）。

全局指定 10 个线程（N=10），主线程负责文件的顺序读入，每读入一个数独记成一个任务（将一个数独存放在全局二维数组里），空闲的线程领取完成任务

（recvAJob 函数），线程任务完成后若有待解决的数独则再去领取完成，在全部数独求解完后将解写入全局的 bool 变量 shutdown 记录任务全做完，退出 5 个子线程。

二、性能测试

程序的性能会受到诸多因素的影响，其中包括软件层面的因素和硬件层面的因素。

本节将分析比较多线程程序与单线程程序的性能差异，以及多线程时几个线程能使性能最优。

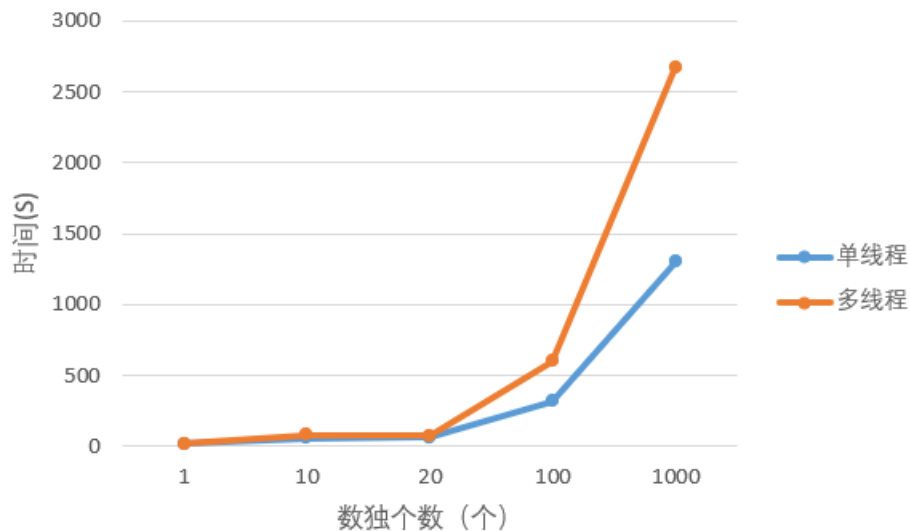
多线程与单线程性能比较

单线程程序只能利用 1 个 CPU 核心，而多线程程序能使 CPU 的多个核心并行运作，因此，多线程能够充分发挥多核 CPU 的优势。

实验提供 6 个不同大小的文件，每个文件分别有数独题：1、10、20、100、1000、10000 个（最大 10000 个数独的文件大小为 800K）。

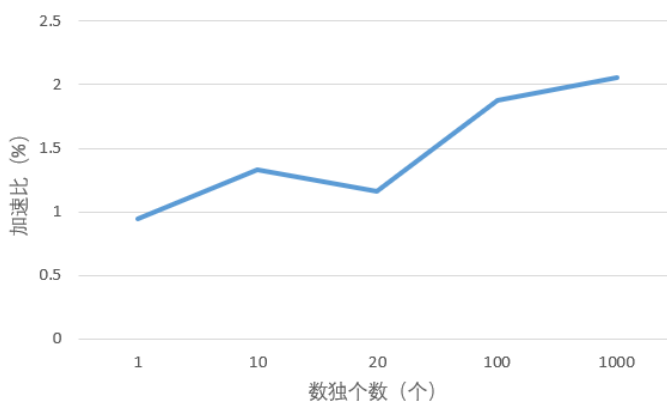
然后分别使用单线程版本和多线程版本（此处定义全局变量为 10 个线程）分别对该文件进行求解测试，比较单线程与多线程的不同耗时，并计算加速比。

如图，具体化单线程和多线程对性能的影响，两条折线：



蓝色线和橙色线分别表示单线程和多线程随着任务数量（数独个数）的变化所需要的时间开销。

从图中可知，刚开始题数为 1，10 和 20 的时候，单线程和多线程的耗时无太大差别，但当数独求解量增多时，单线程和多线程的耗时差别会明显增大，单线程会比多线程耗时增加。



计算加速比=单线程耗时/多线程耗时得上图折线图。可得数独个数越多，加速比越大。

可以大致推测出多线程加速的原因是因为多线程比单线程更好地利用了 CPU 资源，使得多线程在多任务的情况下加速更明显。

但是在任务较少的时候，单线程会反而快于多线程（此处 10 和 20 个数独的情况下单线程比多线程基本快 10-20s），因为数量较小的时候，多线程会花费一部分时间在线程的调度上。

故数独数量为 10 和 20 时，加速比较小，而 1000 和 10000 时加速比能达到接近 2。