

** **HUNAN UNIVERSITY**

云计算技术

实验报告

|  |  |  |
| --- | --- | --- |
|  |  | **对对对队** |
|  |  | 201808010829张继伟（组长） |
|  |  | 201808010824谢正宇（组员） |

目录

实验一

[一、 实验概要 3](#_Toc68731304)

[二、 性能测试 4](#_Toc68731305)

# 实验一

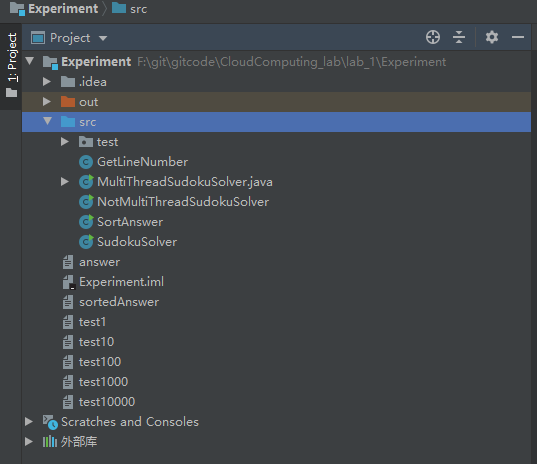
1. 实验概要
   * + 1. **实验环境**

操作系统：Windows10

应用软件：IDEA

计算机语言：JAVA

* + - 1. **代码说明**

****

本工程主要由五个类组成。

GetLineNumber类用于获取样例文件和结果文件的行数；

MultiThreadSudokuSolver类用于测试多线程解决数独的时间；

NotMultiThreadSudokuSolver类用于测试单线程解决数独的时间；

SortAnswer类用于将存在answer文件中乱序的结果排序；

SudokuSolver类用于解决单个数独，并将结果写入到answer文件；

* + - 1. **样例测试说明**

**多线程样例测试**

更换样例测试需更改**MultiThreadSudokuSolver.java**文件第31行

**File file=new File("test10000");**

将“test10000”更改为想要测试的样例文件名

此外需提前将样例文件放入**Experiment**文件夹中

最后重新编译运行**MultiThreadSudokuSolver.java**文件

最终数独结果存入**sortedAnswer**文件中

**单线程样例测试**

更换样例测试需更改**NotMultiThreadSudokuSolver.java**文件第25行

**File file=new File("test10000");**

将“test10000”更改为想要测试的样例文件名

此外需提前将样例文件放入**Experiment**文件夹中

最后重新编译运行**NotMultiThreadSudokuSolver.java**文件

最终数独结果存入**answer**文件中

**注：我们准备了1,10,100,1000,10000五组样例**

1. 性能测试

|  |  |  |  |
| --- | --- | --- | --- |
| 数独大小 | 未排序总时间ns | 排序总时间ns | 单线程时间ns |
| 1 | 12311100 | 14224600 | 8583200 |
| 10 | 67227200 | 78802600 | 47716500 |
| 100 | 156708300 | 262385500 | 215127000 |
| 1000 | 554842500 | 1417746000 | 1247779400 |
| 10000 | 2739074500 | 3816089200 | 10366087100 |

如图，具体化单线程和多线程对性能的影响即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，刚开始数独解题数为1和10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

此外，随着数独解题数的增多，排序与否对实验总结果影响增大，具体是因为I/O操作数目陡增，耗时大幅度增加

同时可以看出，数独解题数越大，多线程耗时与单线程耗时差距增大，即数独个数越多，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，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

**注：本实验结果与电脑性能有关，我们采用的是java的线程池**

**（newCachedThreadPool()）**

**这个线程池创建时总是创建测试电脑可以承受的最大线程，由此可以发现在采用多线程测试时CPU利用率在某一段时间达到99%自由**