

Lab1 test report

1.实验概述

1.1 输入

程序在控制台接收输入的测试文件，测试文件在当前版本的代码的目录下，包括 test1,test10...test1000，文件中包含一个到多个的数独题，并且这些数独按照一定的格式存储。

1.2 输出

程序将测试文件的数独的解按照与输入相应的顺序写到输出文件 outfile 中。

1.3 Sudoku 算法

实验提供了四个算法：BASIC,DANCE,MINA 和 MINAC。本实验选择使用 BASIC 算法。

1.4 实验环境

本次实验有 3 个不同的实验环境，分别是：

(1) yudan: Linux 内核版本为 4.15.0-72-generic; 1GB 内存; CPU 型号为 Intel®Core™i5-7200U [CPU@2.50GHz](#)，共一个物理 CPU，每个物理 CPU 有 4 个物理核心；不使用超线程技术。

(2) tjc: Linux 内核版本为 4.15.0-72-generic; 1GB 内存; CPU 型号为 Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz，共 2 个物理 CPU，每个物理 CPU 有 1 个物理核心；不使用超线程技术。

(3) 服务器: 2GB 内存; 1 个虚拟 CPU

在测试时会说明不同的测试环境

1.5 代码版本

本次实验中使用了三份代码：

- (1) easy_version, 静态分配各个线程任务后各自执行等
- (2) Basic_version, 使用任务队列分配任务等
- (3) Advanced_version, 在 Basic_version 版本基础上, 使用一个专门的线程接受输入, 并且可以接受任何数量的输入文件等。

2.性能测试

本次实验的性能分析将比较

- ① 实验中实现的不同版本的代码在相同的测试环境、不同的输入文件以及同样使用单线程进行求解的性能差别;
- ② 实验中实现的某一版本的代码在相同的测试环境下、相同的文件输入以及不同的线程数量的性能差别;
- ③ 实验中实现的某一代码在不同的测试环境下、相同的文件输入以及相同的线程数量的性能差别。

2.1 不同版本的代码

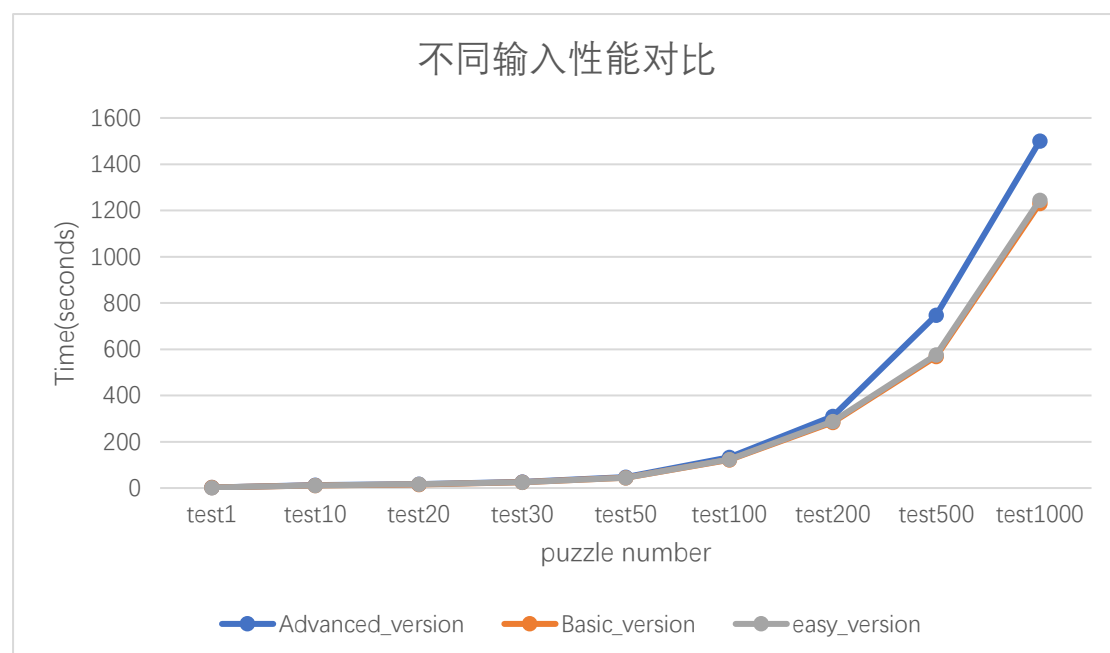
代码版本: Advanced_version, Basic_version 以及 easy_version

输入: 从 test1 到 test1000, 使用单线程求解

测试环境: 服务器

这三个版本的代码实现了相同的功能, 但是在时间上的开销是不同的, 随着问题规模的不断增大, 不同版本的代码时间上的开销的差距会更加明显的显现。本次实验对这些代码进行时间开销上的对比, 分别按照输入文 test1,test10,test20, test30,test50,test100,test200,test500,test100 对代码版本 Advanced_version,

Basic_version 以及 easy_version 进行测试。从下图中可以看出当输入文件的数独数量不断增加，Advanced_version，Basic_version 以及 easy_version 的时间开销差距越来越明显。在 test1000 时 easy_version 比 Basic_version 多花了将近 14 秒的时间，这是因为 basic 比 easy 版本实现代码更多，并且随着输入文件数独问题数量的增多，额外的开销的影响就体现出来了，所以最终会造成上图这种情况。



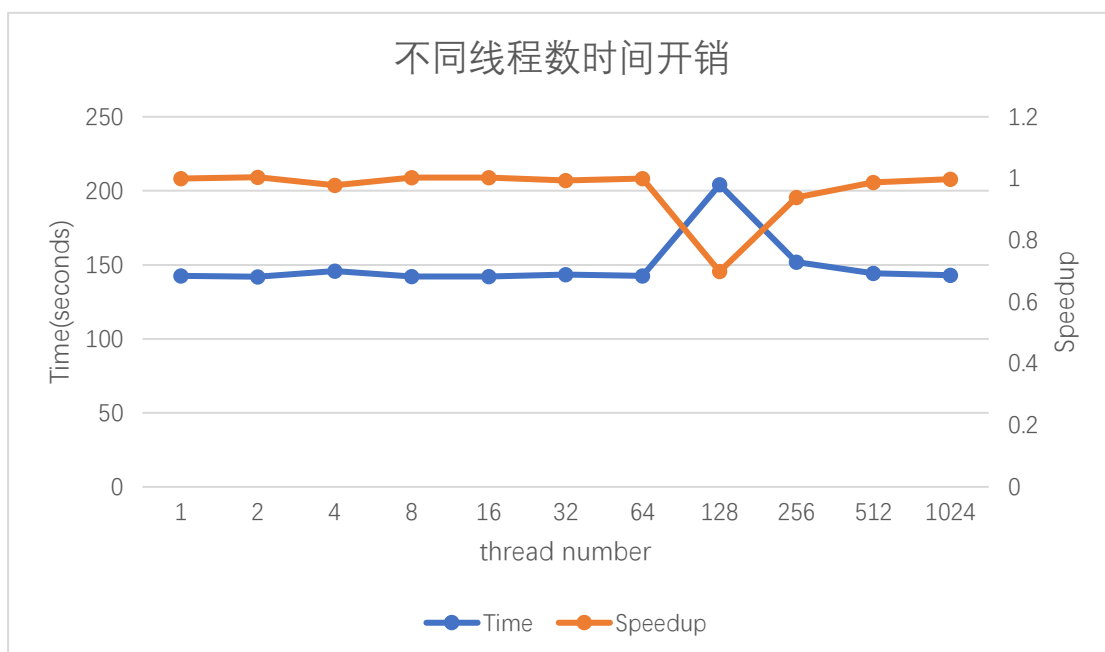
2.2 不同线程数

代码版本为 Basic_version

输入：test100

测试环境：yudan

针对 Basic_version 版本的代码在不同的线程数的使用情况下，测试其时间性能，线程的数量为 1,2,4,8,...1024，使用某个数量的线程对所提供的输入文件 test100 中的数独题进行求解，并且将结果写入输出文件 outfile 中，测量则个过程的时间开销。



2.3 不同硬件环境

实验使用 Basic_version 版本的代码在两个不同的硬件环境下，针对输入文件 test100 进行求解测试，sudoku_solve 线程数量从 1 开始增加，测量性能。

测试环境: tjc、yudan

如下图所示为在不同硬件环境下的测试结果:

