

# Lab1 test report

---

## 1.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。

### 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。

### 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一个文件中。

### 1.3 Sudoku 算法

实验共提供了 4 中不同的 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。实验中选用的是最快的 DANCE 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）。

### 1.5 实验环境

实验中共有 1 个不同的实验环境：

linux 内核版本为 4.15.0-88-generic； 8GB 内存； CPU 型号为Intel(R) Xeon(R)

Gold 6148 CPU @ 2.40GHz，共 1 个物理 CPU；每个物理 CPU 有 4 个物

理核心，共有 4 个物理核心；使用超线程技术，1 个物理核心模拟出 2 个逻辑核心，共有 8 个逻辑核心。

### 1.6代码实现版本

实验中总共使用了四种方法，由DANCE 算法改进的动态多线程分配任务以及静态多线程分配任务的两种代码，以及 回溯算法改进的动态多线程分配任务以及静态多线程分配任务的两种代码。

代码中可通过调节线程数参数改变当前代码中的线程数,也可修改调用的解决算法，在不同的性能分析项目中，会具体给出本版本代码的改动之处。

## 2、性能测试

程序的性能会受到诸多因素的影响，这里主要分析不同算法的代码和如何给线程分配任务对程序性能的影响。

这里将是用两种的算法来分析动态分配任务和静态分配任务对程序性能的影响，

## 2.1 动态多线程与静态多线程性能比较

静态多线程：作业在线程开始处理之前被划分，

优点：低争用开销(每个都有自己的一段输入)，可以并行读取输入更快

缺点：在处理之前需要预先用代码来划分输入作业

负载不平衡(有些线程很快，有些很慢)

比较适合输入作业是成批输入的，每个作业的处理时间相似

动态多线程：作业被动态分配给工作线程

优点：平衡每个线程负载，不需要预先了解输入作业

缺点：实时分配作业时要争用，在线程较少的时候会产生争夺

比较适合动态流式输入作业，线程通信开销很低的作业，还可以使用调度器，调度器是廉价和快速的

## 2.2 首先比较回溯的代码的动态多线程与静态多线程

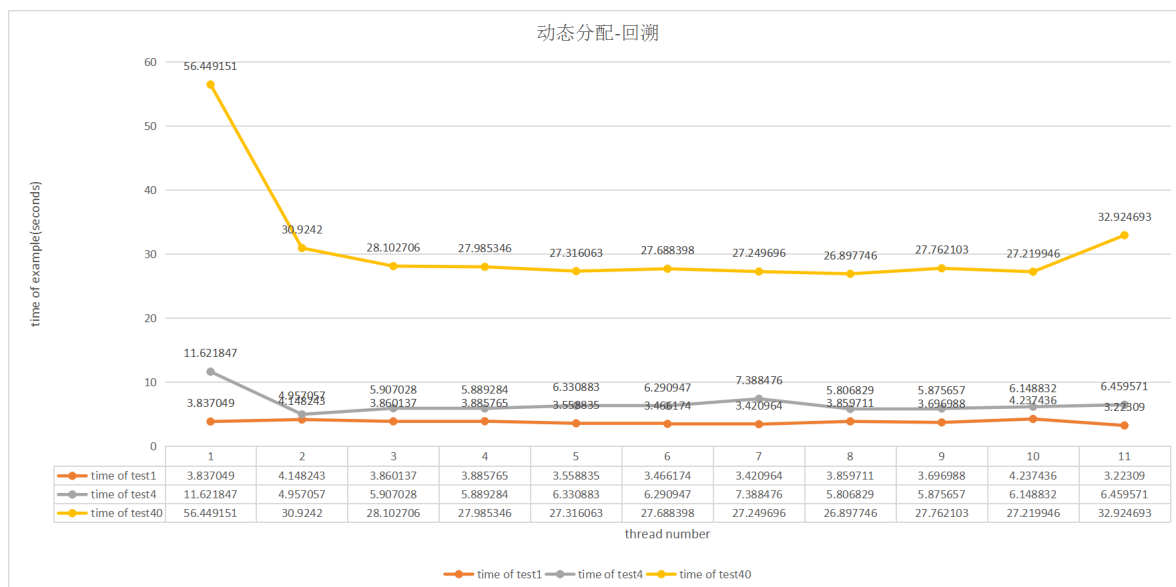


图2-1回溯算法动态多线程

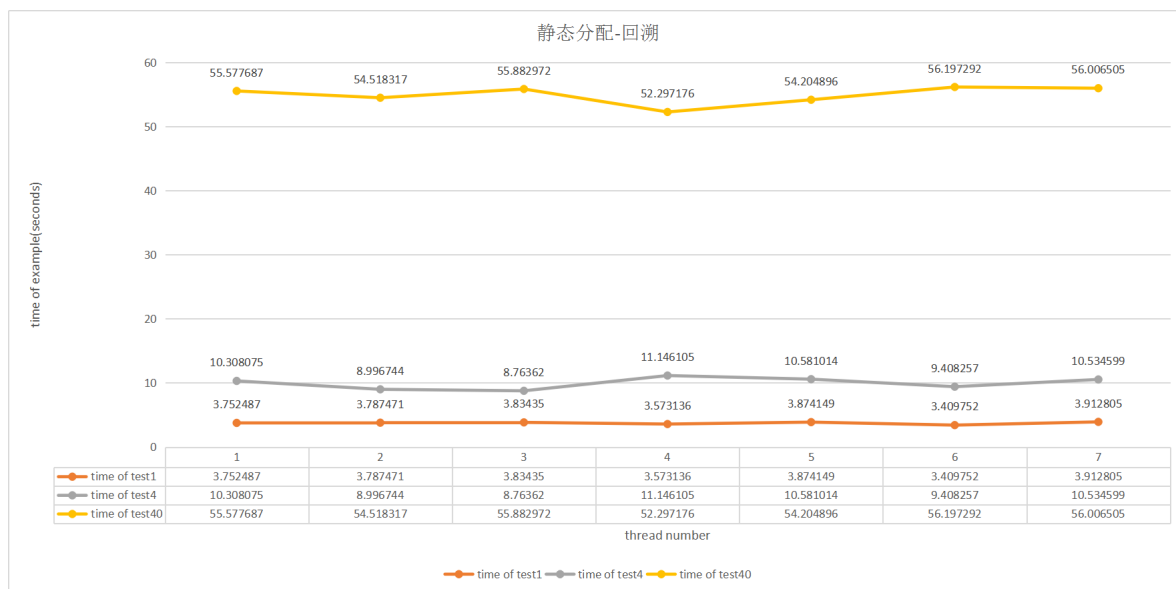


图2-2回溯算法静态多线程

可以明显看出在任务逐渐增多的时候，动态多线程完成时间明显比静态多线程的短，但是在只有一个线程的时候，动态多线程的分配作业的时候发生了争用导致效率下降。

## 2.3比较DANCE的代码的动态多线程与静态多线程

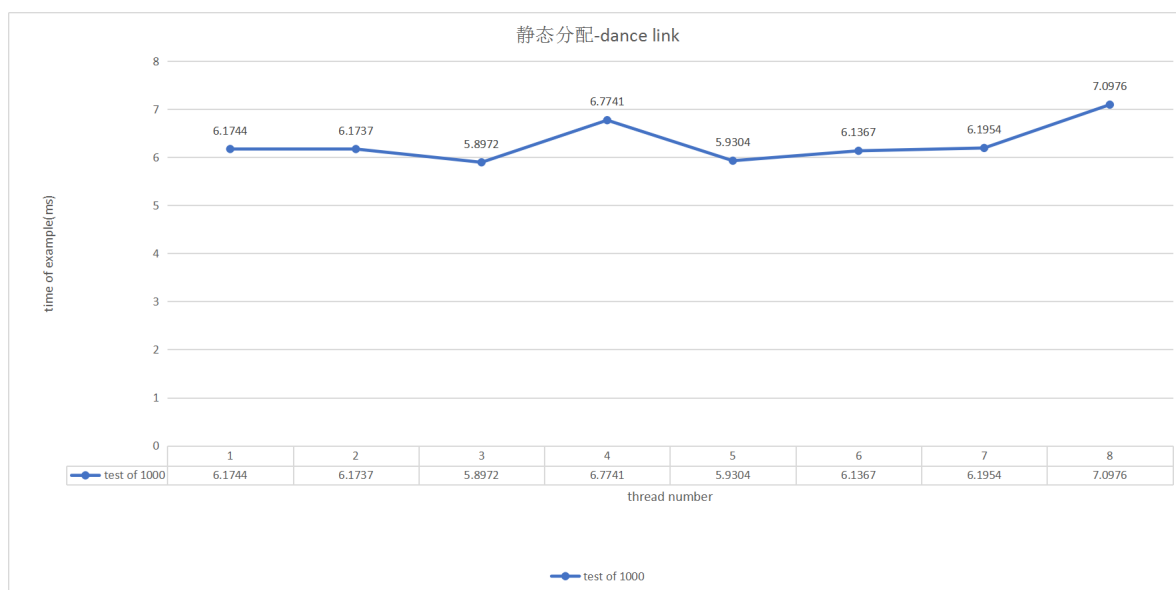


图2-3DANCE算法的静态多线程

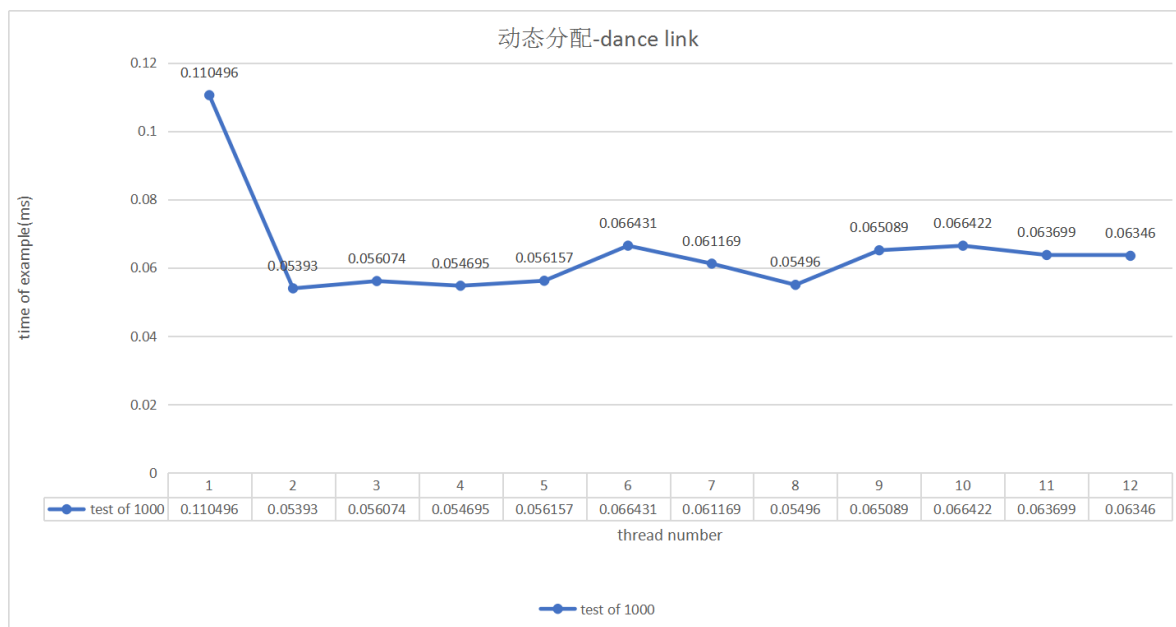


图2-4DANCE算法的动态多线程

可以看到dance算法明显比回溯的算法速度来的更快

动态多线程的分配任务的时候在只有一个线程的时候，还是产生了争用，导致时间过长，但是这个算法在动态分配任务的时候明显比静态分配任务使用的时间少，动态多线程的最慢需要0.110496ms，而静态多线程最慢的需要7.0976ms，可以说在这个算法上，动态多线程明显赢得很轻松。