

并行与分布式作业

多线程计算高维向量之和
第一次作业

姓名：黄炜钊

班级：教务一班

学号：18340066

一、 问题描述

现代处理器如 Intel、ARM、AMD、Power 以及国 产 CPU 如华为 鲲鹏等， 均包含了并行指令集合。

1、请调查这些处理器中的并行指令集， 并选择其中一种进行编程练习， 计算两个各包含 10^6 个整数的向量之和。

2、此外， 现代操作系统为了发挥多核的优势， 支持多线程并行编程模型， 请将问题 1 用多线程的方式实现， 线程实现的语言不限， 可以是 Java， 也可以是 C/C++。

二、 解决方案

对于问题 1:

我用的是 Intel 的内部指令集——AVX 进行编程练习。根据参考实现的内容， 我选择单精度浮点数 float 类型进行实验测试， 两个向量则由随机数生成。

对于问题 2:

我选择的是 pthread 多线程执行， 定义的线程数量为 8。同样地， 选择单精度浮点数 float 类型进行实验测试， 两个向量由随机数生成。

三、 实验结果

首先是问题 1 的实验结果， 如下表所示：

	1	2	3	4	5	平均值
AVX 实 现	5041μs	6052μs	5984μs	7032μs	5697μs	5961.2μs

for 循环	11124μ	11047μ	14960μ	9975μs	12303 μ	11881.8μs
实现	s	s	s		s	

因此，我们可以得到加速比为： $11881.8/5961.2 \approx 1.99$ 。这个加速比看上去并不高。因此我将向量的维度扩展至 10^7 （只需要修改开头的#define vec_length 的值即可），并得到加速比约等于 2.18，在此便不再将所得数据一一列举出来了。

接着是问题 2 的实验结果，如下表所示：

	1	2	3	4	5	平均值
多线程	30181 μ	35144 μ	22039 μ	30783 μ	32359 μ	30101.2
实现	s	s	s	s	s	μs
for 循环	54382 μ	56420 μ	54651 μ	57161 μ	62562 μ	57035.2
实现	s	s	s	s	s	μs

因此，我们可以得到加速比为： $57035.2/30101.2 \approx 1.89$ 。

注意，此处的得到的所有数据是将向量的维度设置为 10^7 、线程数设置为 8 得到的。同样地，我也将向量的维度和线程数进行了多次组合的修改得到了更多的数据，在此便不再一一赘述。

四、遇到的问题及解决方法

在本次作业中，我遇到的问题不算多。大致上可以分为以下两个部分：

- （一）环境的配置问题：本次作业我分别选择了 AVX 和 Pthread，同时选择的 ide 是 DevC++ 而非手动编译。

因此在环境的配置方面我刚开始是找不到方向的。在这个方面我是通过请教其他同学进行解决的。

- (二) 实验结果方面：正如很多同学在群里面询问那样，我也出现了很多次不符合我心中预期的实验结果：如单线程的运算时间比多线程的运算时间短；使用了 avx 向量加法的加速比小于 1 等等。后来在经过思考和查阅资料后这些疑问大多都得到了解决。例如，加速比小于 1 可能是没有关掉 -O 优化导致的；多线程的运算时间长可能是因为向量的维度不够高，也可能是因为多线程里面的其他调用产生的额外开销导致的。