并行与分布式作业

“作业名字”

第二次作业

姓名：黄炜钊

班级：教务一班

学号：18340066

1. 问题描述
2. 分别采用不同的算法（非分布式算法）例如一般算法、分治算法和Strassen算法等计算计 算矩阵两个300x300的矩阵乘积，并通过Perf工具分别观察cache miss、CPI、 mem\_load 等性能指标。
3. Consider a memory system with a level 1 cache of 32 KB and DRAM of 512 MB with the processor operating at 1 GHz. The latency to L1 cache is one cycle and the latency to DRAM is 100 cycles. In each memory cycle, the processor fetches four words (cache line size is four words). What is the peak achievable performance of a dot product of two vectors? Note: Where necessary, assume an optimal cache placement policy.
4. Now consider the problem of multiplying a dense matrix with a vector using a two-loop dot-product formulation. The matrix is of dimension 4K x 4K. (Each row of the matrix takes 16 KB of storage.) What is the peak achievable performance of this technique using a two- loop dot-product based matrix-vector product?
5. 解决方案

对于问题一，我按照题目要求使用了3种方法进行矩阵的运算，然后使用perf工具观察性能指标。

对于问题三，因为第一个迭代创建一个缓存未命中，接下来的3次迭代只访问缓存数据中的a[i]和b[i]，所以每4次迭代就有2次未命中a[i]和b[i]。所以，4次迭代持续的内存延迟周期为：2\*100=200.在4次迭代中，我们的操作数为4\*2=8.因此，每200ns，就有8/2\*10-6=40Mflops。

对于问题四，在预热阶段之后，向量保存在cache中，不会再创建未命中cache。类似地，未命中是每4N次操作中出现一次，是可以忽略的。因此，每4次迭代，就有一个cache未命中。4次迭代持续100个周期，执行4\*2=8次操作。所以每100ns的8ops就是80Glops。

1. 实验结果

对于问题一的实验结果，如下表所示

|  |  |  |  |
| --- | --- | --- | --- |
| 性能指标 | 传统算法 | 分治算法 | Strassen算法 |
| cycles | 2.251GHZ | 2.372GHz | 2.372GHz |
| Task-clock | 175.97msec | 1715.56msec | 12214.36msec |
| Context-switches | 0.147K/sec | 0.171K/sec | 0.144K/sec |
| Cpu-migrations | 0.006K/sec | 0.002K/sec | 0.002K/sec |
| Page-faults | 0.002M/sec | 0.478K/sec | 0.149M/sec |
| branches | 153.358M/sec | 513.874M/sec | 1021.153M/sec |
| Branches-misses | 0.35% of all branches | 0.01% of all branches | 0.15% of all branches |

以上的表格数据采用的是多次取值然后算平均得到的结果。（Tips：传统算法矩阵大小为300\*300，分治和strassen算法为512\*512）

1. 遇到的问题及解决方法

本次作业遇到的最大的问题就是我的电脑跑不了perf，在尝试了多种方法后，最终还是选择让别的同学帮我跑了一下。其次遇到的问题是分治算法和strassen算法的理解，我在理解并应用这些算法上花费了一定的时间，最后才勉强有所领会，以后我还是需要多加接触和练习这方面。