并行程序设计与算法第一次作业-答案

March 10, 2024

1 简答题

习题 1

为求全局总和例子中的 my_first_i 和 my_last_i 推导一个公式。需要注意的是：在循环中，应该给各个核分配数目大致相同的计算元素。(提示：先考虑 n 能被 p 整除的情况)。

解答：

```
quotient = n / p;
remainder = n % p;
if (my_rank < remainder) {
    my_n_count = quotient + 1;
    my_first_i = my_rank * my_n_count;
}
else {
    my_n_count = quotient;
    my_first_i = my_rank * my_n_count + remainder;
}
my_last_i = my_first_i + my_n_count;
```

习题 2

(1) 解释局部性原理

(2) 在以下的代码中，存在何种局部性?

```
float z[10000];
float sum = 0.0;
for (int i = 0;i < 2000;i++)
    sum += z[i];
```

解答：

(1) 空间局部性：访问一个位置的数据后，访问其附近的数据。时间局部性：访问一个数据后，在不远的将来继续访问该数据。

(2) 对 z 数组的访问存在空间局部性；对 sum 变量和 i 变量的访问存在时间局部性。

习题 3

(1) 当 CPU 将数据写入缓存时，缓存中的值可能与主存中的值不一致，有哪两种解决策略？请阐述。
(2) cache 映射的方式有哪三种？请阐述。

解答：

(1) 有写直达和写回策略：(i) 写直达策略在将数据写入 cache 时，同时修改主存中的数据；(ii) 写回策略只修改 cache 中的数据，并将 cache 中的数据标记为 dirty 行；当改行数据被换出时，将新值写入到主存中
(2) cache 的映射方式有三种：全相连映射、直接映射、组相连映射。全相连映射：新行可以放置在 cache 中的任意行中；直接映射：一行只能被放置到唯一的 cache 行中；组相连映射：一个 cahce 组有 n 行，每个 cache 行可以被放置在唯一的 cache 组的任意行中。

习题 4

在冯 · 诺依曼系统中加入缓存和虚拟内存改变了它作为 SISD 系统的类型吗？如果加入流水线呢？多发射或硬件多线程呢？

解答：

加入缓存和虚拟内存既不会改变一次性可执行指令的数量也不会改变一次性可执行的数据量。但是，复杂的系统可以提供一些并发性：例如，当 cache 未命中时，CPU 可能会尝试执行一些不涉及不可用数据或指令的指令，我们把这样子的系统称为 MIMD 系统：加载和存储指令可以与其他指令同时执行。

若加入流水线，因为我们可以将流水线视为将一条复杂指令用于多个数据对象，所以我们把这样子的系统称为 SIMD 系统。

若加入多发射或者硬件多线程，因为他们尝试对不同的数据项应用不同的指令，所以我们把这样子的系统称为 MIMD 系统。

2 计算题

习题 5

在下列情况中，推导公式求出 0 号核执行接收与加法操作的次数 (假设一共有 p 个核)。
a. 在课本 1.3 节的例子中，第一种计算全局总和的算法 (0 号核作为 master 核)。
b. 在课本 1.3 节的例子中，第二种计算全局总和的算法 (树形结构)。
c. 制作一张表来比较这两种算法在总核数是 2、4、8、...、1024 时，0 号核执行的接收与加法操作的次数。

解答：

a. 最初代码要执行 $p-1$ 次接收与加法操作，其中 p 为核的数量。
b. 树形结构 0 号核要进行 $\lceil log_2 p \rceil$ (向上取整) 次接收核加法操作。

c. 表如下：

p	初始算法	树形算法
2	1	1
4	3	2
8	7	3
16	15	4
32	31	5
64	63	6
128	127	7
256	255	8
512	511	9
1024	1023	10

习题 6

回顾之前一个从缓存读取二维数组的示例 (课本 2.2.3 的实例)。请问一个更大矩阵和一个更大的缓存是如何影响两对嵌套循环的性能的? 如果 MAX =8，缓存可以存储 4 个缓存行，情况又会是怎样的? 在第一对嵌套循环中对 A 的读操作，会导致发生多少次失效? 第二对嵌套循环中的失效次数又是多少?

解答：

当缓存行行数不变时，矩阵规模越大，发生缺失次数越多，性能越低；当矩阵规模不变时，缓存行行数越多，发生缺失次数越少，性能越高。

当 MAX=8，且有 4 个缓存行时，采用第一对嵌套循环，因 1 个缓存行只能装载 4 个元素，因此需要 2 个缓存行才能装完 A 矩阵一行元素，故 A 矩阵每访问一行会发生 2 次 cache 缺失，总计 $2 \times 8 = 16$ 次；若采用第二对嵌套循环，则每访问一个元素均会发生 1 次 cache 缺失，总计 $8 \times 8 = 64$ 次。