# 一、题目分析

1.1测量Cache容量

利用Cache原理，可以在内存中读取一块连续的数据，数据大小逐渐增大，测量平均读取速度。由Cache的作用可知，当所读连续数据大小超过Cache大小，则发生Cache缺失，平均访问时间会发生较大幅度的增大，找到该突变点，即可测量Cache容量大小。

1.2 测量Cache块大小

CPU与Cache之间的数据交换是以字为单位，而Cache与主存之间的数据交换是以块为单位，一个块是由若干字组成，是定长的。也就是说，当某个字节访问缺失时，读进来的是一个块的大小。故可以设置递增的间隔读取一段连续存储空间，当间隔超过Cache块大小时，会出现Cache缺失，则访问时间也会发生较大幅度的增加，找到该突变点，即可测量Cache的块大小。

1.3 测量Cache相联度

假设Cache是N路组相联，只需要连续访问能够映射到同一个Cache组的超过N个不同的区域，当访问第N+1个区域时，对应Cache组内的块均已装载满，则会发生Cache缺失，同样地，访问时间也会发生较大幅度的增加，找到该突变点，即可测量Cache的块大小。

1.4 矩阵乘法优化

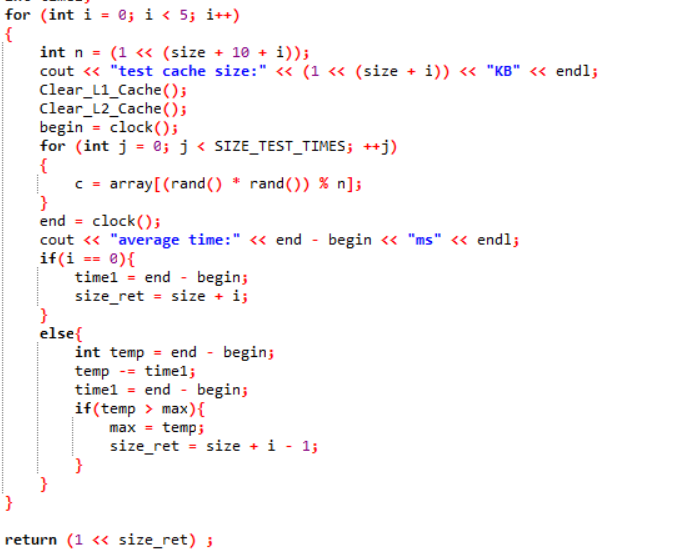
一般矩阵乘法按照c[i][j] += a[i][k] × b[k][j]进行计算，显然对b数组的访问不是按行访问，可以利用空间局部性，进行顺序调整，提升性能。

# 二、设计思路与关键实现

2.1测量Cache容量

思路：如果一个数组能够完全存放在L1 DCache中，那么在一个循环里不停的访问数组，第一遍循环数组就会从主存被搬到了L1 DCache中，以后每次循环就会直接从L1 DCache中读取数组数据，这样速度就会很快。如果一个数组的大小超过了L1 DCache的大小，那么整个数组就不能被完整地存放在L1 DCache中。此时，存在某种访问方式，使得数组的循环访问每次都是缺失的，都必须从L2 Cache中找，即出现了题目分析中的突变点。因而，可以逐步增加访问一个数组区域大小，测量其访问时间，找到该突变点。

实现：采用随机数的方法，用取余数的方法控制访问数组的区域，每次测量的访问次数保持相同，寻找两次访问时间之差最大的即为突变点，返回即可。

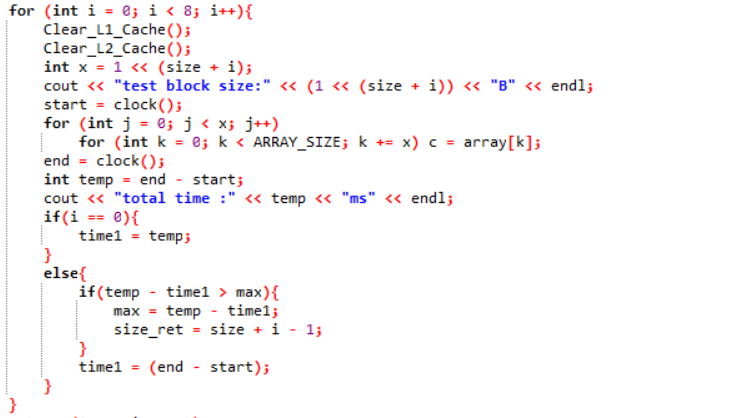


2.2测量Cache块大小

思路：对于数组访问，如果是连续访问，第一个字节缺失的话，一个块就被存放在了Cache中，以后Cache块顺序访问的字节就命中了，因此命中率就会很高。 如果访问是间断的，对数组间隔顺序访问，命中率就会降低，平均访问延迟增大。当某次间隔达到一定大小，即超过Cache块大小，有可能造成每次都缺失的最坏情况。

  结合数组的大小和间隔的大小，依次增加间隔的大小和数组访问的大小，会使得访问同样次数的时间变大。然后处于跳变的值即为Cache块大小。

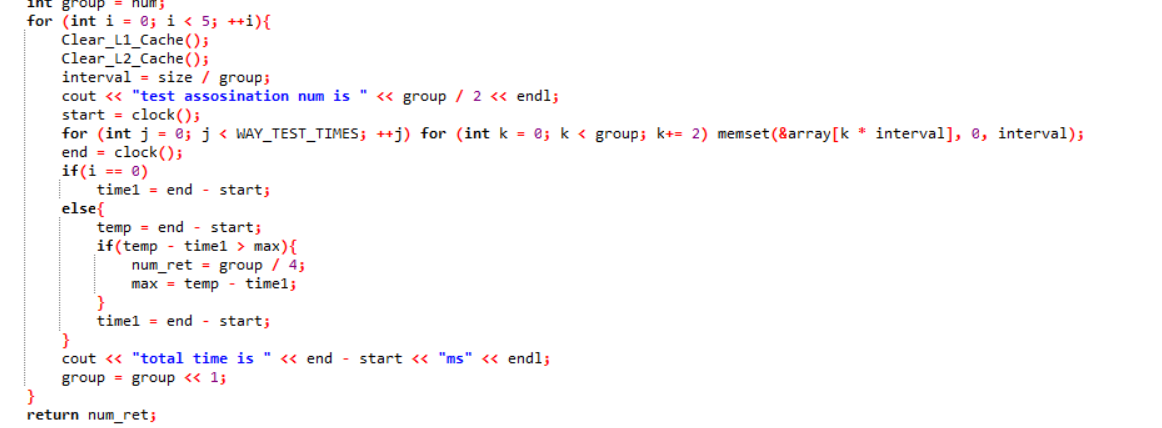
实现：从初值开始，每次增加访问间隔，用for循环访问，两次for循环保证访问的总次数相同，从而计算时间，找到两次访问间隔最大的即为突变点，返回即可。



2.3测量Cache相联度

思路：通过访问一个大小为Cache容量二倍的数组，来测试相联度。如题目分析所述，只需要将数组分组，从而能够使映射到同一组的数据块大于相联度，然后连续访问，寻找突变点即可。如当前测试的是2路组相联，由于数组大小是Cache二倍，将数组平均分为八块，则一、三、五、七块对应Cache的同一组，依次访问，则Cache会不停的换入换出，访问时间增加。因此可以逐渐增加数组的块数，当某一次访问时间变慢时，前一次就是相联度的值。

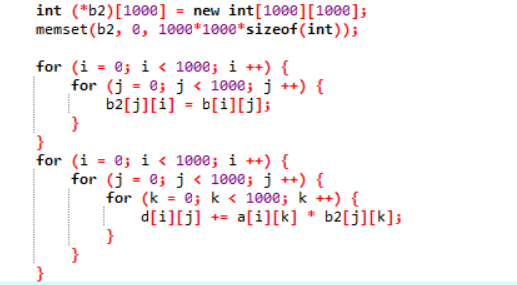
实现：如图所示，其中group为数组分组数，interval即为组间间隔。不停地遍历每一分组统计时间找出突变点即可。



2.4矩阵乘法优化

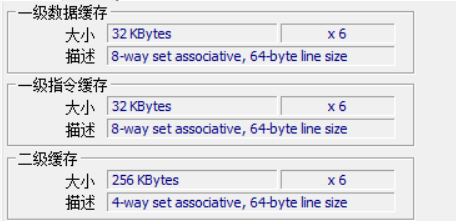
思路：按题目分析所述，按原先公式对数组b的访问不是按行访问，故先对数组b进行转置，记为b2，随后用b2参与矩阵的乘法运算

实现：如图所示

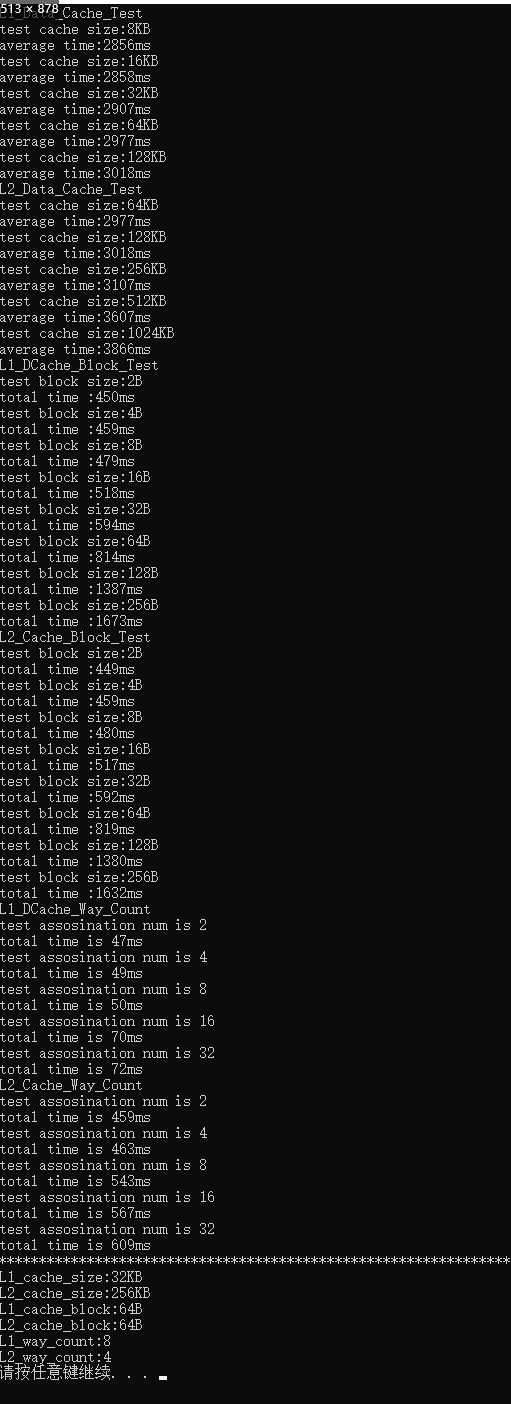


# 三、测试结果及分析

机器Cache参数如图所示：



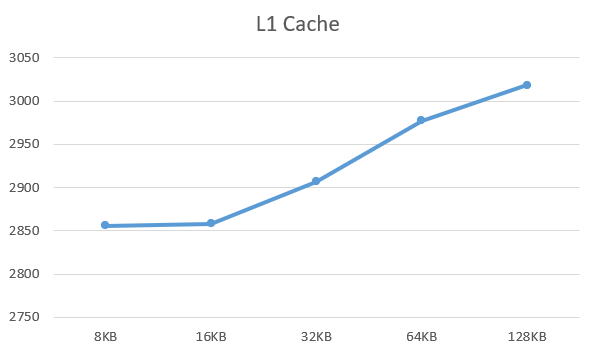
程序测量结果如图所示：



可据此绘制折线图：

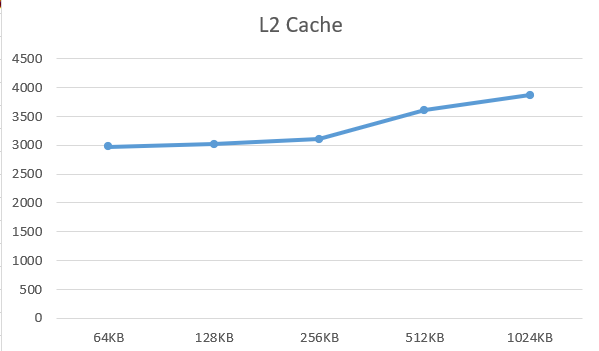
3.1测量Cache容量

L1 Cache访问时间随容量增加的折线图如图所示（纵轴单位为ms，横轴单位为KB）：



可以发现，突变点为32KB到64KB，故L1Cache容量大小为32KB

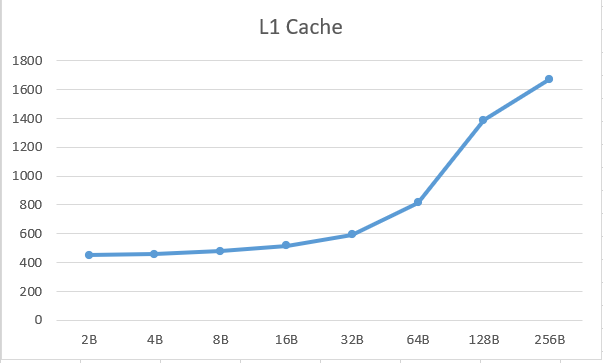
同理，绘制L2 Cache的访问时间随容量增加的折线图：



可以发现，突变点为256KB到512KB，故L2 Cache容量为256KB

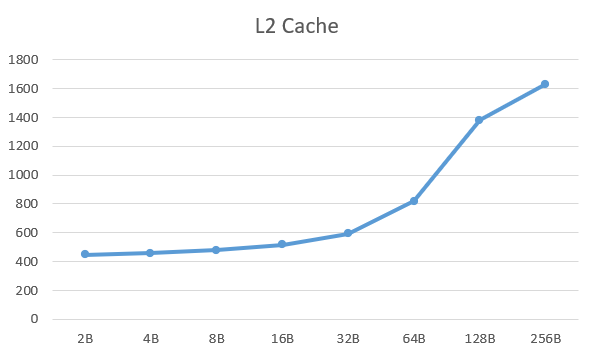
3.2测量Cache块大小

根据实验结果将访问时间随L1 Cache块大小的变化绘制成折线图，如图所示，其中横轴为相联度，纵轴为访问时间（ms）：



可以发现，突变点为64B到128B，故L1 Cache的块大小为64B

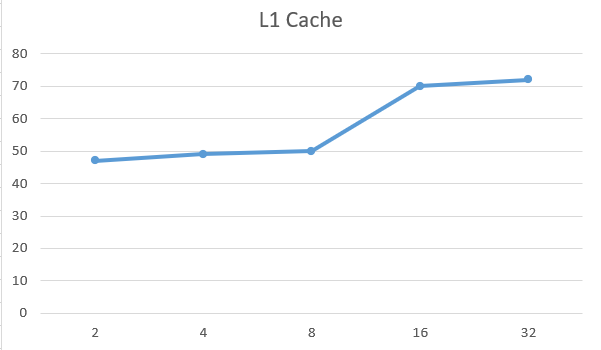
同理画出L2Cache访问时间随块大小变化的折线图：



可以发现，突变点也是64B到128B，故L2 Cache的块大小为64B

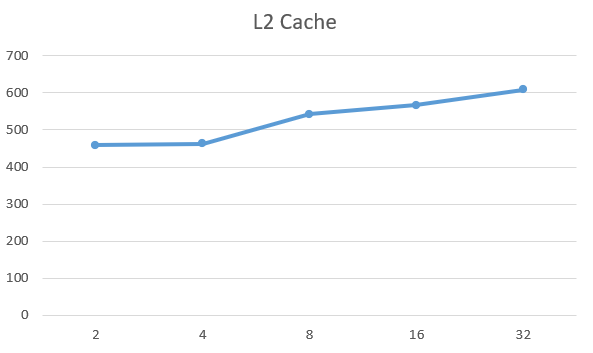
3.3测量Cache相联度

根据实验结果，将访问时间随L1 Cache相联度的变化绘制成折线图，如图所示，其中横轴为相联度，纵轴为访问时间（ms）：



可以发现，突变点为相联度从8到16，L1 Cache的相联度为8

同理画出L2 Cache的折线图：



突变点为相联度为4变成8，故L2 Cache的相联度为4.

3.4矩阵乘法优化

优化前后矩阵乘法所花费时间结果如图所示



可以发现，对数据按行访问的空间局部性效果更好，能够减少矩阵乘法所用的时间。