存储层次分析及程序优化

1190303311 王志军

1. 题目分析

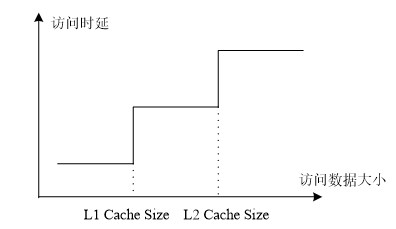
本次实验要求通过程序测试实验机的Cache各项参数，包括L1数据cache和L2cache的容量、块大小和相联度，最后还需要对矩阵乘法根据cache的结构进行优化；

1. cache容量

测量Cache容量最直接有效的办法是不断地在内存中读取一块连续的数据，数据块的大小可以从1kB一直到16MB，然后观察平均读取速度。当数据块大小超过L1 Cache后，会出现Cache读缺失，平均访问时延会有一个突然增加的过程；同理，在数据块大小超过L2 Cache后，平均访问时延也会有一个突然增加的过程。测量Cache块大小，主要是利用L1 DCache、L2 Cache 和主存（或L3 Cache）之间的速度差异；

测量过程中要保证取平均读取速度；

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，比如是L1 DCache的两倍，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；而L2 Cache的速度低于L1 DCache的速度，使得这个数组的访问延迟明显比其它小数组访问延迟要大。



1. 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。

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。

1. cache相联度

测量相联度可通过访问一个2倍于Cache大小的数组来实现。假设Cache采用的是2路组相联的结构。如果将数组平均分为4块，则访问数组的第一块和第三块时，由于Cache是2路组相联，数组第一块和第三块中的数据块会分别被映射和存放到同一个Cache组里。此时，循环访问数组的第一块和第三块不会发生不命中。但如果将数组平均分为8块，并且访问数组的第一块、第三块、第五块和第七块，这时虽然第一、三、五、七块中的数据块分别被映射到同一个Cache组里，但一个Cache组只能容纳两个数据块，因此不停地循环访问第一、三、五、七块时，会造成Cache块不断的被替换出去，使得访存时间明显增加。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。

1. 设计思路
2. cache容量测量

经过初步探索，发现cache测量时使用时间并不明显，因此此处选择循环访问cache固定次数，从8KB大小开始，每次增加一倍大小，访问数组内容，这样总的访问次数是一样的，访问策略是产生一定范围内的随机数然后对数组大小取余，这样保证每次只能访问到当前大小内的元素。最后记录时间，求出突变点，突变前的大小就是cache大小。

1. cache块大小测量

访问整个数组，从步长为1开始，每次步长增加一倍，刚开始时，由于cache每次都是读取一个块，会有较多的块命中，从而时间较短，到步长增加到cache块的大小时，每次访问cache都会造成不命中，这样就会在访问时间看到明显的突变，突变时的步长就是cache块的大小。

1. cache相联度测量

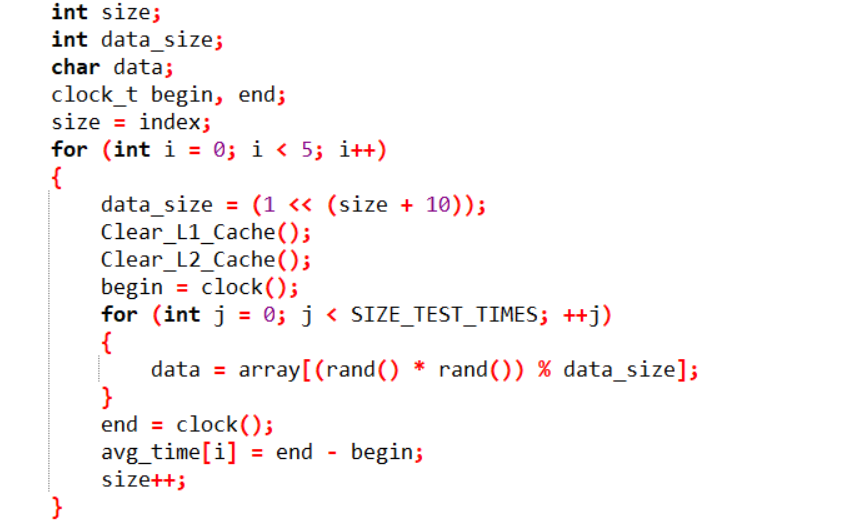
对于相联度，由于cache容量已知，每次访问都是cache二倍大小的数组，从把数组分为两份开始，每次份数增加一倍，依次访问一份，当达到二倍于相联度时，会导致每个组中的cache都被转载，冲突不命中次数增加，访问时间突变，突变前一次就是相联度。

1. 矩阵优化

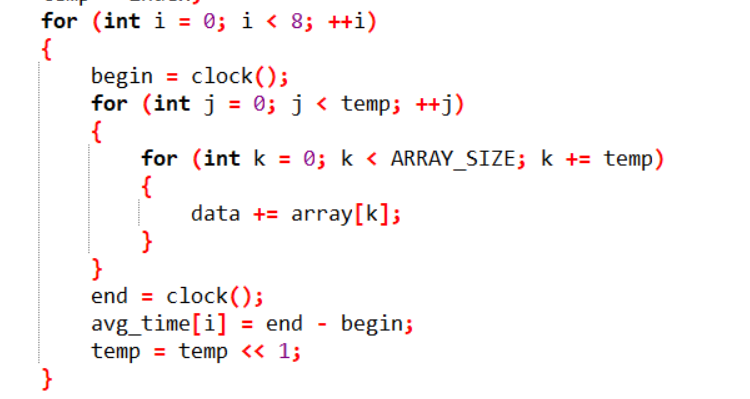
矩阵乘法时间久的原因主要是b矩阵访问是按列进行，不符合cache的读取策略，这里简单的提前将b矩阵进行转置。

1. 关键实现

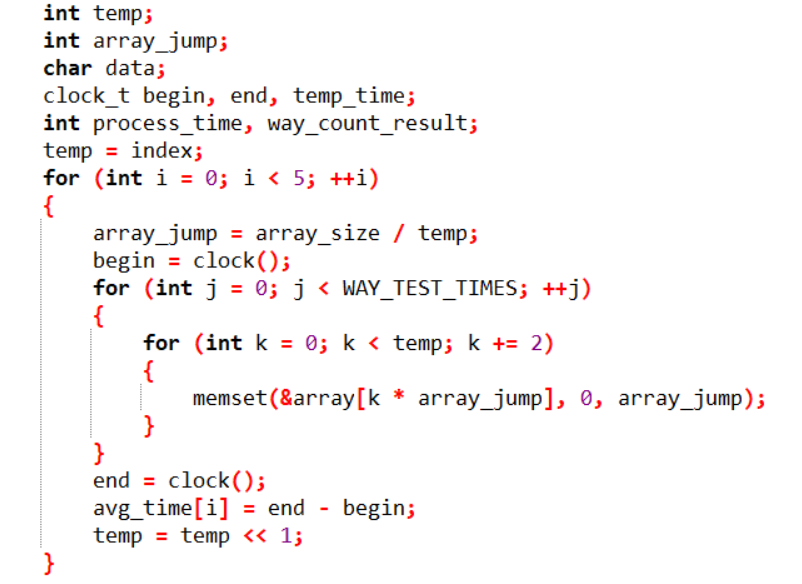
1、cache容量的测量，为了保证每次测量都访问同样次数的数组，设置每次访问次数固定，同时每次通过产生随机数并对相应数组大小取余来限制访问范围：



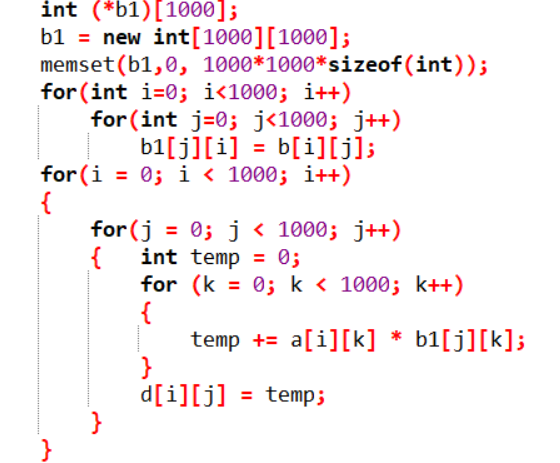
2、测量块大小时，每次按步长访问



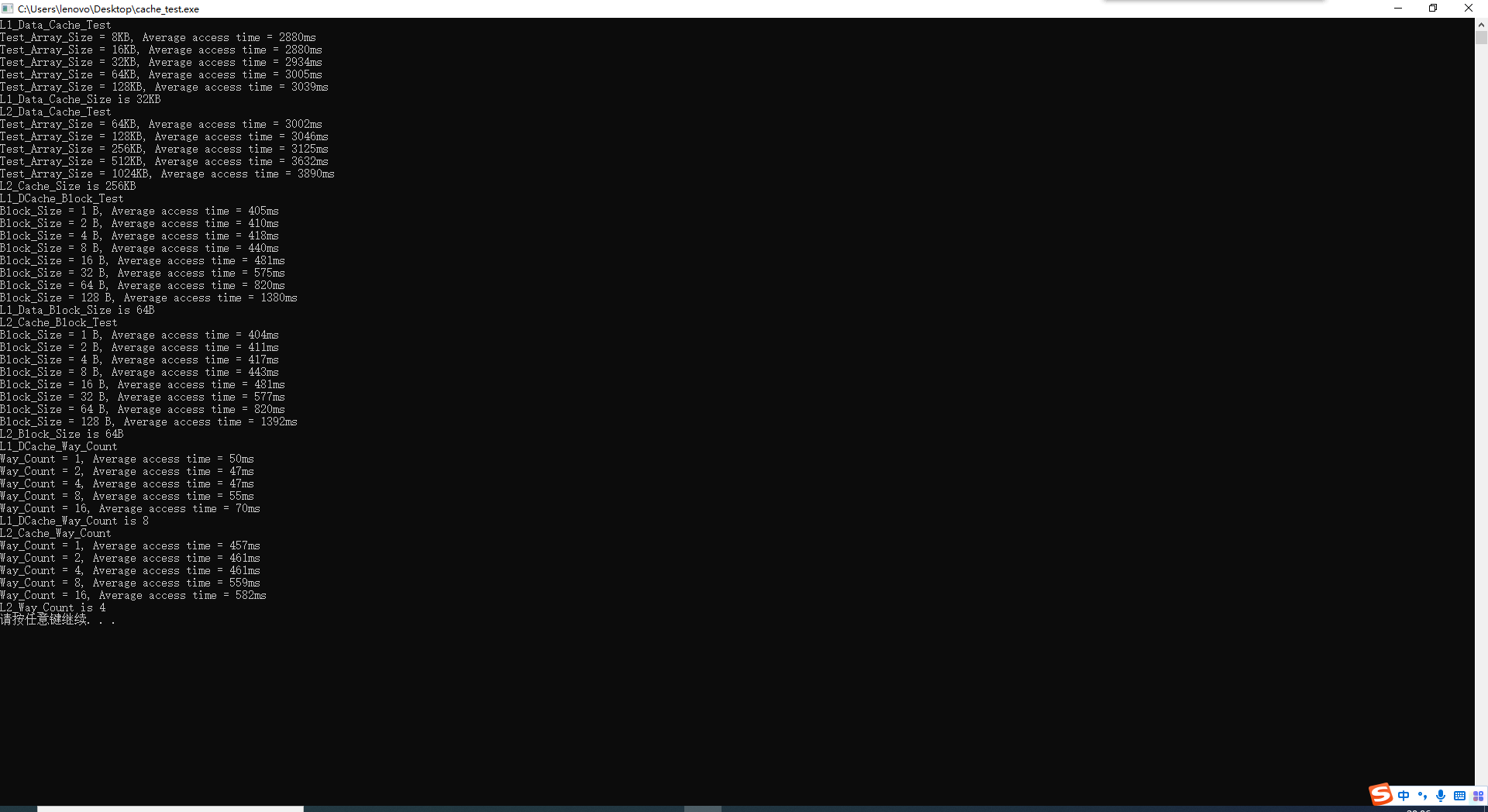
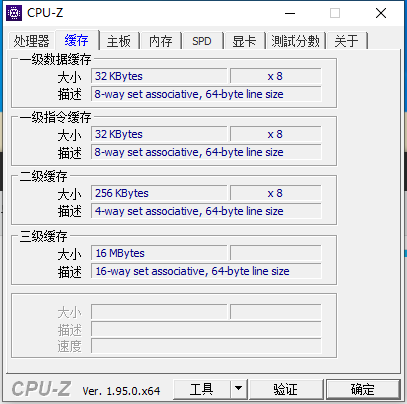
3、cache相联度的测量，由于需要将数组平分为多组，导致每次访问次数不一样，无法比较，这里使用memset函数，依次set对应组大小个字节，保证每轮访问的总次数是一样的：



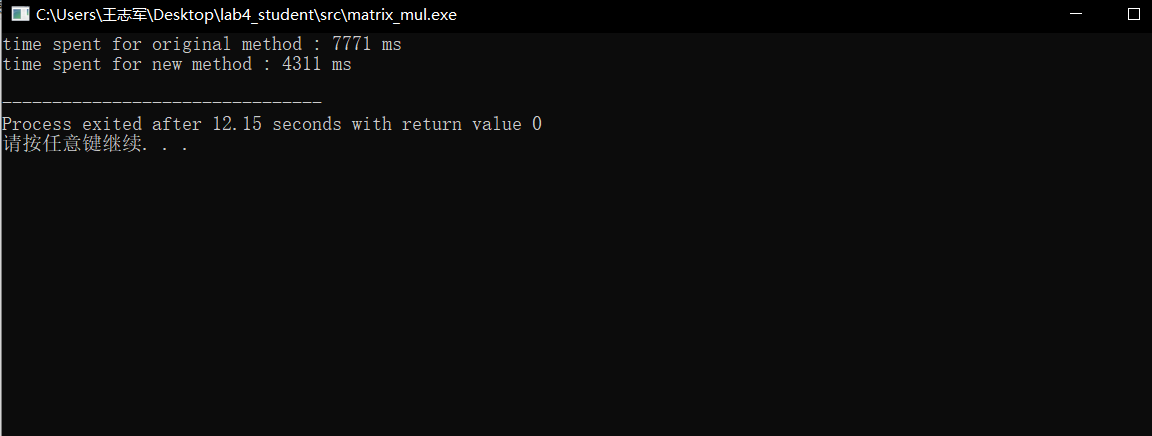
1. 矩阵优化，先把矩阵b进行转置：



1. 测试结果及分析



上图是CPU-Z测量的结果和程序测量的结果，可以发现各项数据都能匹配，并且相关位置平均访问时间的变化也比较明显；



矩阵优化的效果如上，可见直接先转置b矩阵也是有效果的。