

实验报告

开课学期： 2022秋季

课程名称： 计算机体系结构(实验)

实验名称：实验4：层次存储系统分析

学生班级： 计算机6班

学生学号： 200110619

学生姓名： 梁鑫嵘

实验与创新实践教育中心印制

2022年11月

# 1. 题目分析

*回顾课上讲解的Cache参数测量方法，谈谈你对其原理的理解。若采用PPT之外的测量方法，介绍其原理并分析其正确性。*

本次实验中需要按一定顺序访问内存，并测量平均访问时间，通过增大参数时突然变大的平均访问时间确定系统硬件层次储存系统结构。

其中关于Cache测量，实验中实现了三种参数的测量：Cache大小、Cache块大小、Cache组相连度。由于访存平均时间=Cache命中率\*平均命中访问时间+Cache缺失率\*平均缺失访问时间，所以当Cache在某参数增大过程中平均访存时间突然增大，则可以确定此处突然发生很多Cache缺失，进而推导出此处有Cache参数的变化，再依据访存顺序确定引起Cache参数变化的原因。

# 2. 设计与实现

*\*注：若做了附加题，除了提交相应代码，还需按照指导书要求，在报告中添加相应的分析、设计过程和运行结果截图和对比等内容，否则不加分。*

## 2.1 方案设计

*详细阐述各测量函数的关键实现，建议结合算法流程图等图形化工具来辅助描述。*

本实验中以命令行输出条的方式直观地一边测试一边绘图，每个测试通过一个函数和参数的组合进行测试，并返回当前测试的时间。不断更改参数并记录时间，然后在上一层函数中绘制出来。

对于Cache大小的测量，需要按一定顺序访问一定大小区间内的内存，并不断扩大访问区间大小，当平均访存时间突然变大则可确定当前大小是L1或者L2大小。为减小预取、牺牲Cache等硬件优化的影响，需要采用写内存的方式访存，并且访存间隔不能太小。

for (register int k = 0; k < loop\_const; k++) {  
 for (register uint32\_t index = 1; index < size; index += 2048) {  
 array[index] = index + 1;  
 access++;  
 }  
 }

以上为测量Cache大小的核心代码，采用写访存方式，并每次写距离长度是2048。

对于Cache块大小的测量，步骤和上面类似，不过这次需要改变的参数是写地址的步长。

for (register uint32\_t index = 0; index < size; index += m) {  
 array[index] = index + 1;  
 access++;  
}

以上代码通过固定size=1 MiB，改变步长m，使得当m大于等于块大小的时候每次访存都会缺失，于是能够观测到Cache块大小。

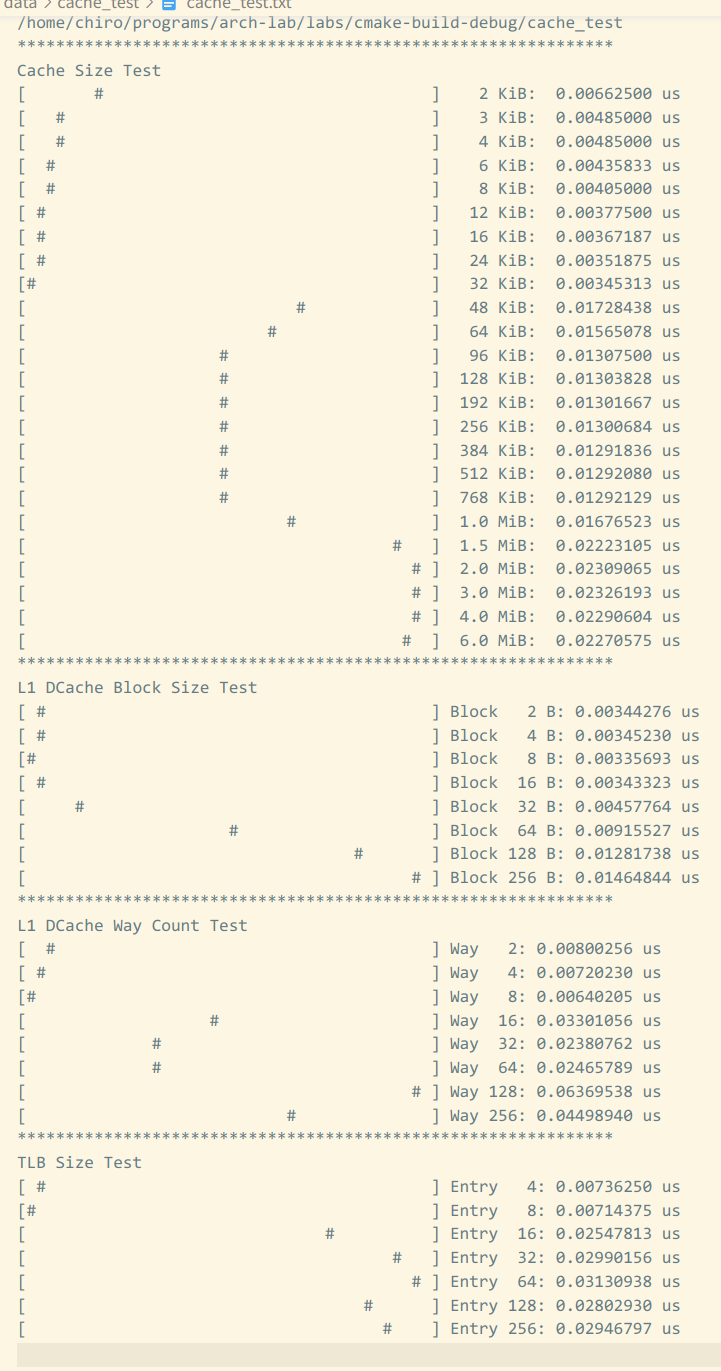
对于Cache相连度的测量，步骤和上面也类似，不过需要改变的是将两倍大Cache数组分作2^n块的参数n，得到剧变的way=n-1。

size\_t size = **KiB**(48) << 1;  
size\_t blk = size >> w;  
gettimeofday(&tp[0], nullptr);  
for (register int i = 1; i < loop\_const / 8; i++) {  
 for (register int k = 1; k < (1 << w); k += 2) {  
 size\_t s = k \* blk;  
 for (register uint32\_t index = s; index < s + blk; index += 1) {  
 array[index] = index + 1;  
 access++;  
 }  
 }  
}  
gettimeofday(&tp[1], nullptr);

以上为测量Cache Way的核心代码，使用大小为size\*2的数组空间，记当前w=n，则每个小块大小为blk=size/(2^n)=size>>w，并每次访问奇数个块，记录时间。

for (register int k = 0; k < loop\_const / 128; k++) {  
 for (register uint32\_t index = 0; index < (1 << w); index += 1) {  
 array[(index << 15) & (ARRAY\_SIZE - 1)] = index + 1;  
 access++;  
 }  
}

以上代码是另一种测量Cache相连度的方法，通过每次以Cache步长访问内存，在访问(1<<way)次后应当造成访存缺失。由于本机的L1 Cache并不是2^N大小，所以这样的方法不一定管用，虽然实际测试起来效果貌似不错，但是应该没有参考价值。



## 2.2 实验结果及分析

*绘制测量各参数时的平均访存时间折线图，并分析实验结果。如果测试结果不明显，试分析原因。*

*如果进行了优化，需给出优化前后的结果对比截图及文字分析（如果优化没有效果，也需分析原因）。*

实验结果如右图所示。

在48KiB、64B、

16Way、16Entry

的曲线处发生了剧烈的上升，而本机L1Cache参数为：48KiB、64B、

12Way、16Entry。具体参数见附录。

可以看出实验效果较好，能较准确地得到当前机器参数。

# 3. 回答问题

## 3.1 为了进一步提高访存性能，现代处理器往往利用时间局部性和空间局部性，将所访问数据块相邻的数据块预取到Cache中。试分析数据预取将给实验结果带来什么影响？

数据预取将预测下一次访存的地址并提前将地址取到Cache中，如果访存时使用的是读操作，则很可能本来应该缺失的地址被预取预测而没有缺失。

## 3.2 试分析Cache对测量TLB造成何种影响？如何排除Cache的干扰？

在访存测量过程中，我们无法判断当前时间突然增大是由于Cache缺失率突然增大造成的还是TLB缺失造成的。为了排除Cache的干扰，需要在测量过程中尽量使得每次访存都Cache缺失。

## 3.3 测量TLB前，是否需要遍历访问一次全部空间？为什么？

需要，需要清除TLB内部的数据。

## 3.4 要写出高效的程序，必须对底层硬件有足够的了解。请谈谈如何利用Cache或TLB提高程序性能。

程序应该尽可能地应用硬件优化，让软件尽可能体现程序的局部性特征，才能让硬件优化更加适配软件程序。为了提高程序的局部性，可以申请更小更局部的数组内存、优化操作系统调度等。

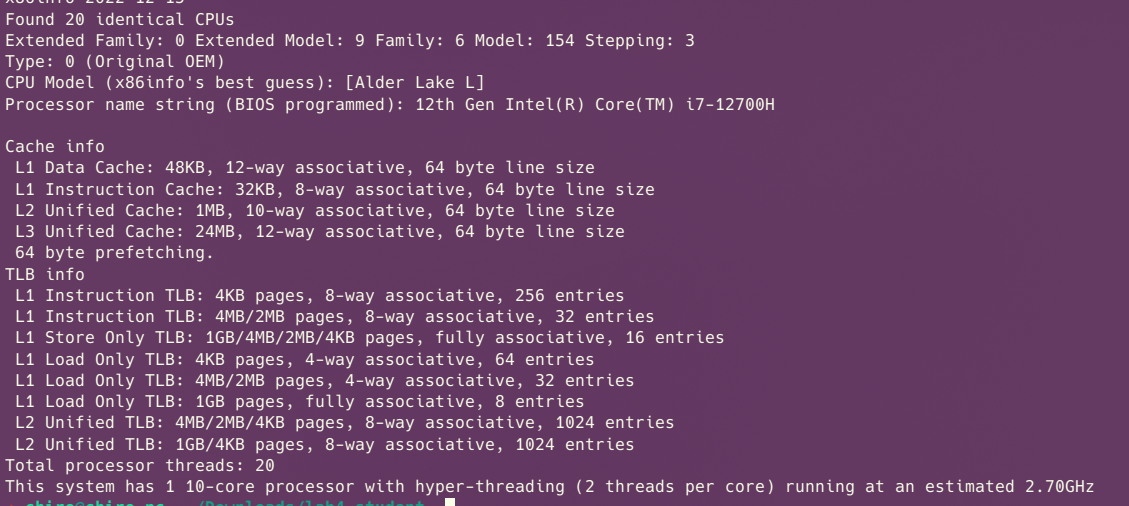
# 4. 总结

*请谈谈学习本课程实验的收获和反思，并提出合理的意见和建议。*

在本次实验中本人丰富了体系结构相关的知识体系，了解了更多体系结构的实验方法。

由于每个人的计算机的体系结构有很大差别，测量Cache和TLB等实验不一定适用于所有电脑，有可能需要固定实验平台。

附录：

本人的实验环境比较复杂…