

# 计算机组织与体系结构 Lab 3.1

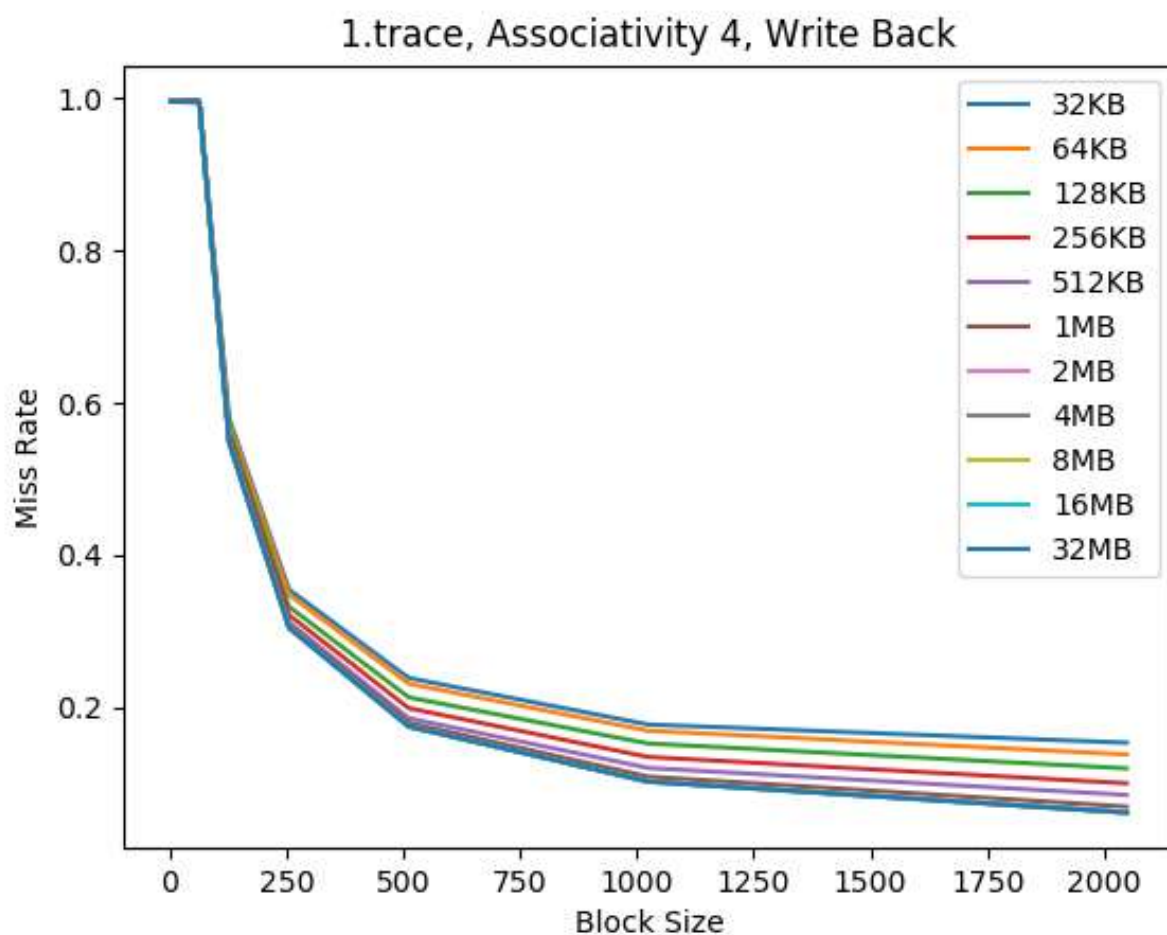
薛犇 1500012752

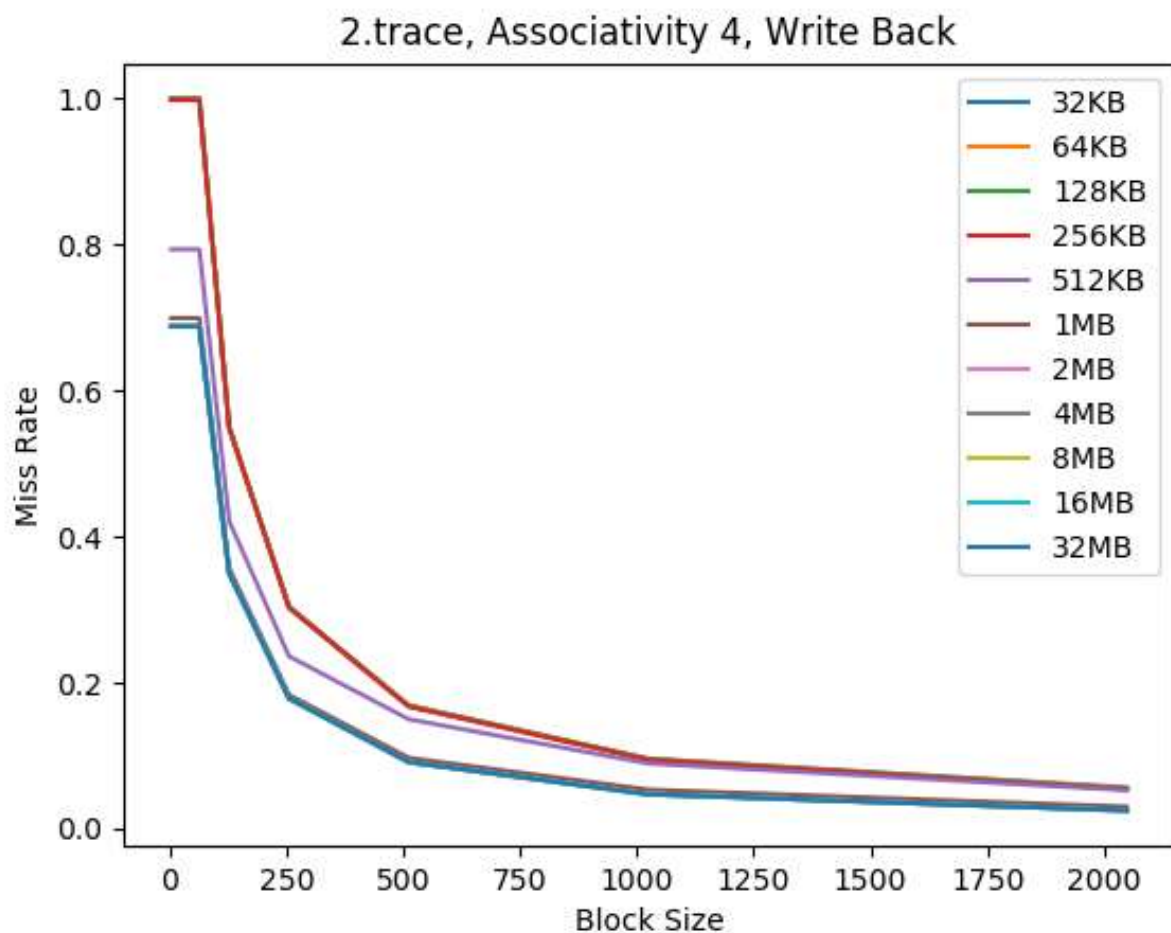
## 1. 测试 trace 文件

(1) 测试在不同的 **Cache Size**条件下， **Miss Rate** 随 **Block Size** 的变化趋势

数据见附件的lab\_data.xlsx文件，折线图如下

在关联度为4， 写回策略下测试





可以看出，随着block size的增加，当达到某个特定值时，miss rate会显著下降，原因可能是block size的大小已经达到了程序操作的地址范围。再之后，miss rate的下降速率会减慢。

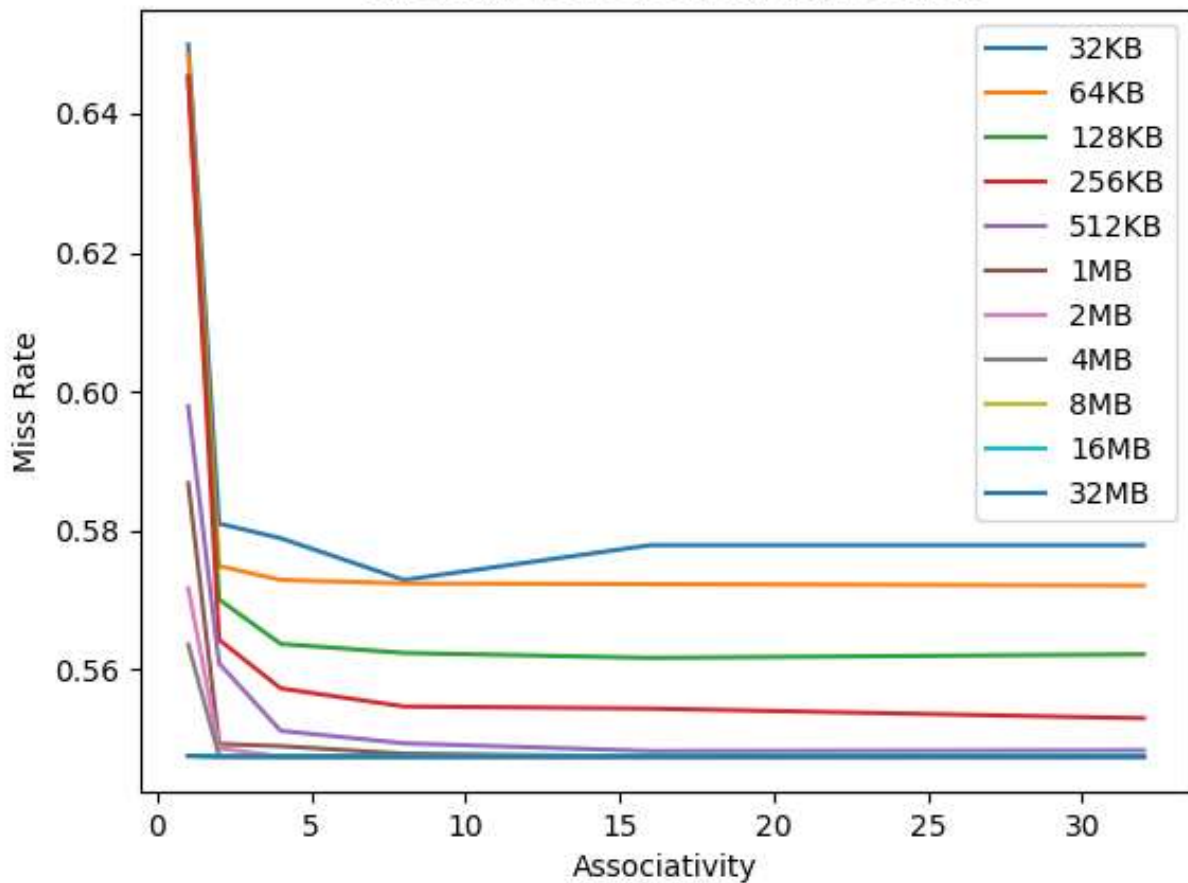
不同的cache size之间，cache size越大，miss rate越低

## (2) 测试在不同的 Cache Size条件下，Miss Rate 随 Associativity 的变化趋势

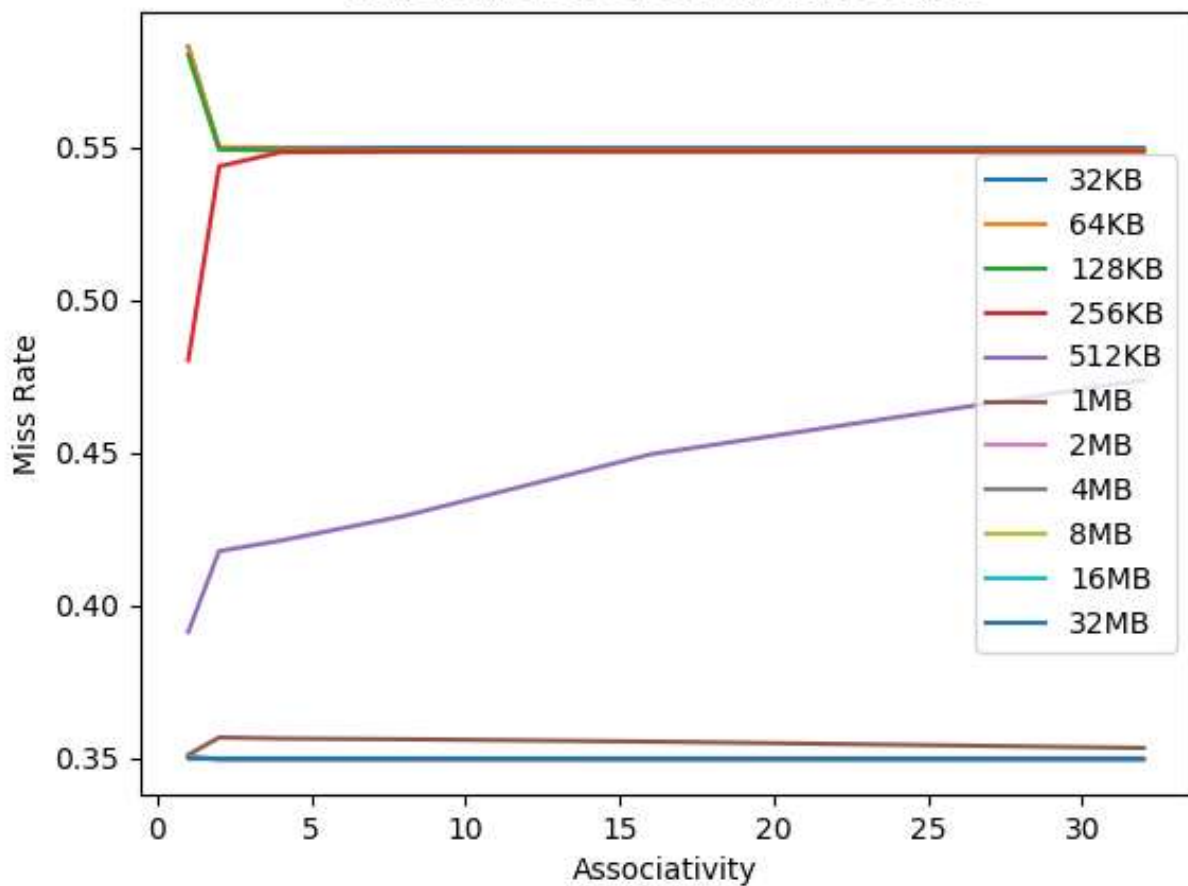
数据见附件的lab\_data.xlsx文件，折线图如下

在块大小128 Byte，写回策略下测试

### 1.trace, Block Size 128, Write Back



## 2.trace, Block Size 128, Write Back



(3) 测试各种写测略的总访问延迟

以 1.trace 文件在cache size = 32KB, block size = 128Bytes, associativity =4的情况下，各种写策略的访问时间

	write ba ck	write through + write allo cate	write through + non-allo cate
l1 cache access tim e(ns)	43170	41890	41980
memory access tim e(ns)	1207950	1220750	1219850
total access time	1249840	1263920	1261830

可以看出，适用了写回策略之后，在cache中的访存时间增加，访存时间减少，带来了总访存时间减少的效果。

对比写分配与写不分配策略，写分配策略有略多的访存时间，原因可能在于，trace文件中存在对某两片索引位相同但是标记位不同的冲突区域反复读写的操作，如果采用写不分配，就会使写操作不影响冲突的区域

2. 与 Lab2.2 的流水线模拟器联调

测试结果如下：

Testing Prog	CPI in lab3.1	CPI in lab2.2
test1	23.55	2.45
test2	19.45	2.45
test3	8.42	2.45
test4	8.76	2.51
test5	9.66	4.09
test6	12.54	2.44
test7	9.46	2.51
test8	17.77	2.97
test9	14.53	2.12
test10	19.33	2.25

对比Lab2.2，可以看出，CPI明显增加，主要原因在于访存的时钟周期变长。由于访存的周期惩罚变成100个周期，所以按理来说CPI会增幅数十倍，但是只增加了几倍的原因在于，我们引入了cache的机

制，由于测试样例的数据地址空间、指令地址空间都比较连续，所以cache的加速效果非常明显。