

天津大学计算机学院

计算机工程系

计算机体系结构

## 实验一：Cache存储体系设计

通用Cache仿真模型及单级Cache仿真器

# 实验报告

3013216096

梁振铎

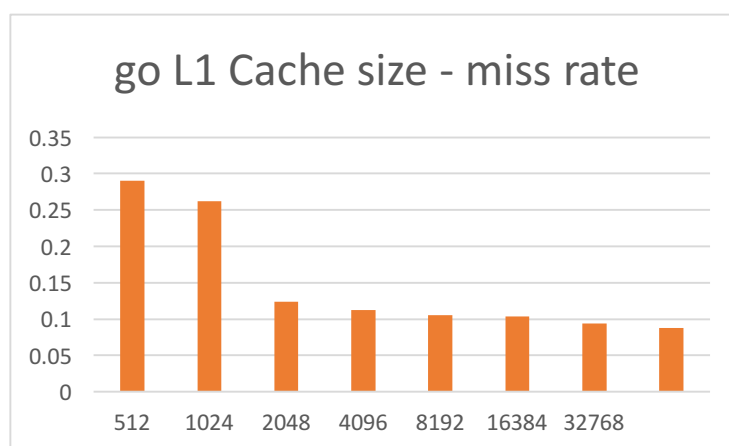
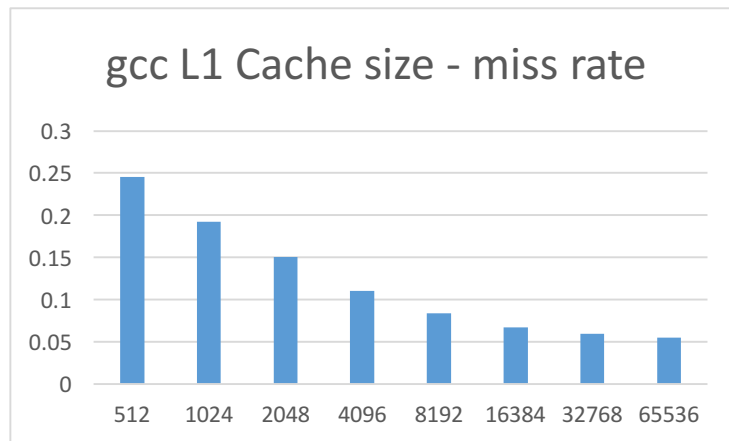
2015 年 12 月 1 日

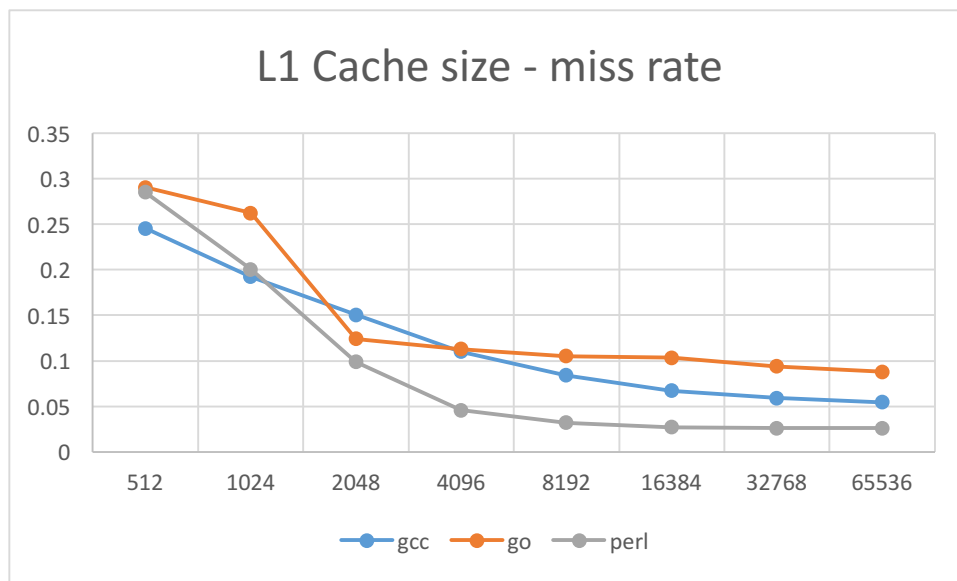
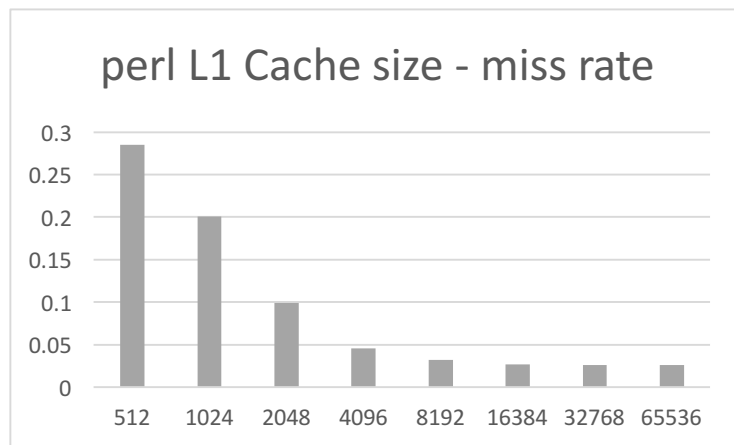
## 一、分析讨论各种体系结构参数对于 Cache 缺失率的影响

### 1、L1 Cache size vs. miss rate

块大小为 16B，相联度为 1，替换策略为 LRU，写策略为 WBWA 时：

Cache/B	gcc	go	perl
512	0.2454	0.2904	0.285
1024	0.1922	0.2623	0.2004
2048	0.1506	0.1242	0.0989
4096	0.1102	0.1128	0.0458
8192	0.0839	0.105	0.0321
16384	0.0672	0.1033	0.027
32768	0.0591	0.094	0.0261
65536	0.0546	0.088	0.0261





结论：缺失率随 Cache 空间的增加而降低，但 Cache 空间达到一定限度时对缺失率的影响将不明显。

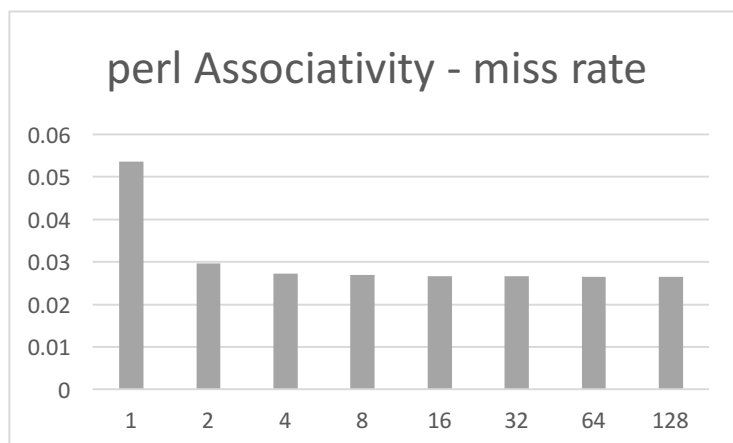
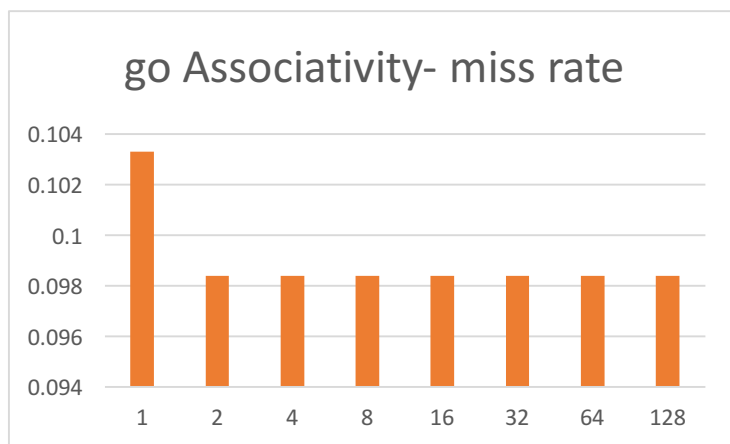
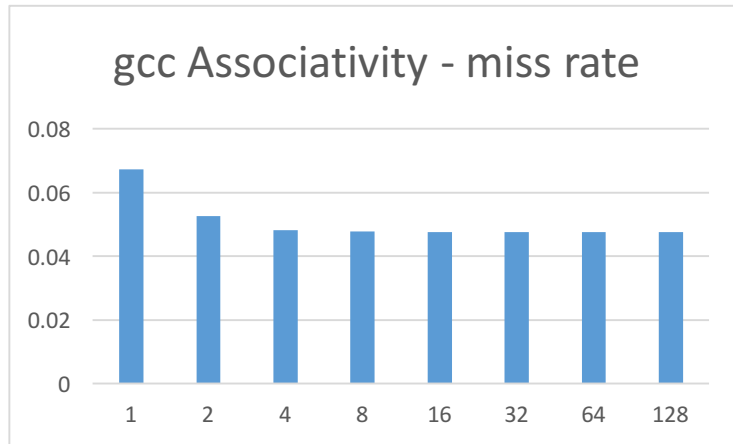
## 2、Associativity vs. miss rate

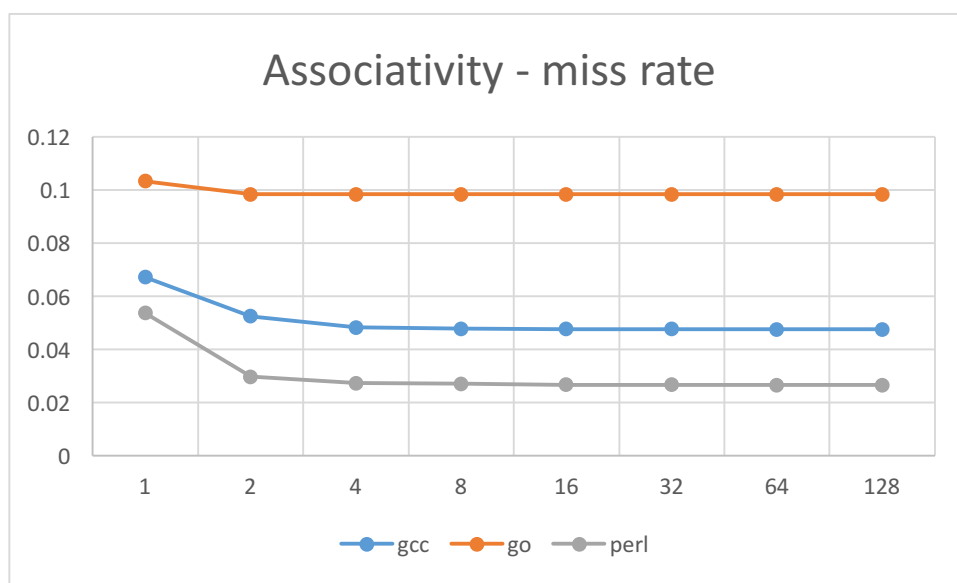
块大小为 16B，Cache 大小为 16384B，替换策略为 LRU，写策略为 WBWA

时：

Assoc	gcc	go	perl
1	0.0672	0.1033	0.0536
2	0.0525	0.0984	0.0297
4	0.0482	0.0984	0.0273
8	0.0477	0.0984	0.027
16	0.0476	0.0984	0.0267

32	0.0476	0.0984	0.0266
64	0.0475	0.0984	0.0265
128	0.0475	0.0984	0.0265



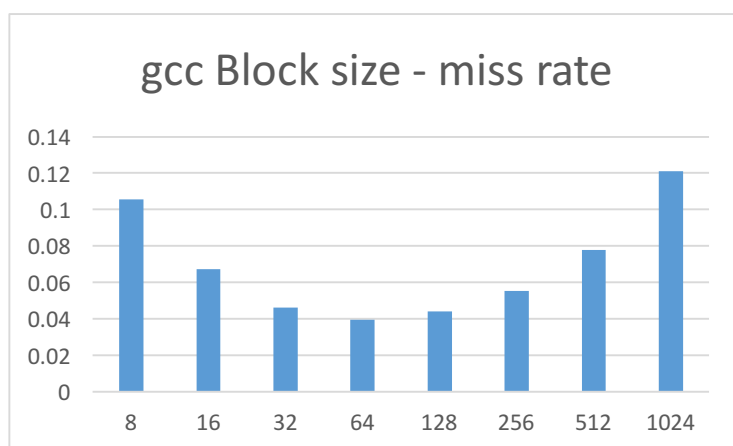


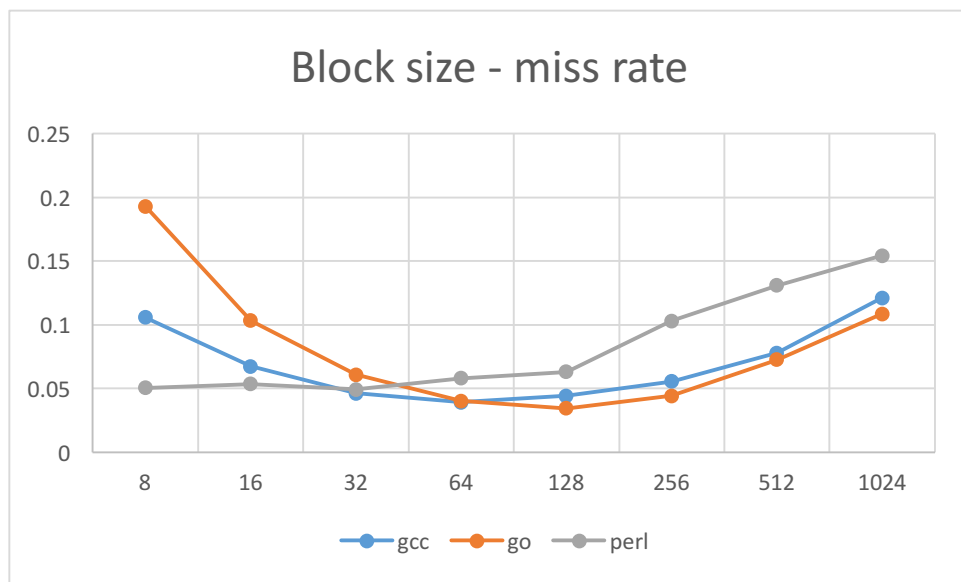
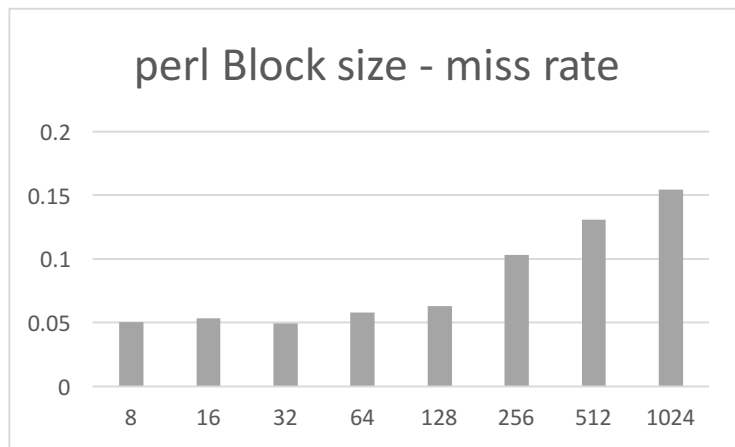
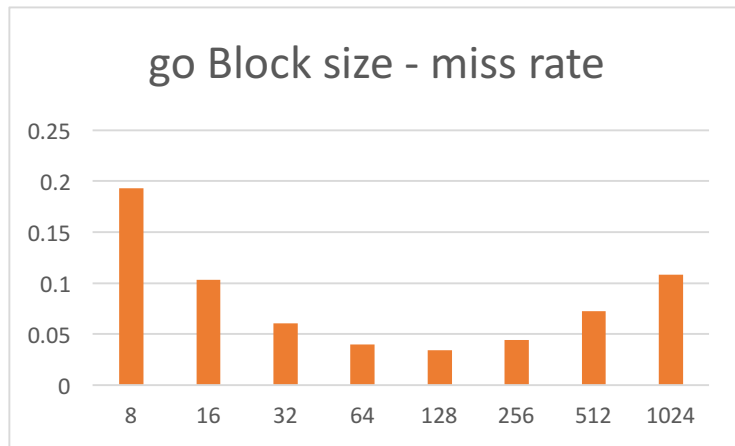
结论：缺失率随相联度的增加而降低，但 Cache 将不再影响缺失率。

### 3、Block size vs. miss rate

Cache 大小为 16384B，相联度为 1，替换策略为 LRU，写策略为 WBWA 时：

Block/B	gcc	go	perl
8	0.1057	0.193	0.0505
16	0.0672	0.1033	0.0536
32	0.0461	0.0607	0.0492
64	0.0393	0.0401	0.0579
128	0.0441	0.0343	0.0631
256	0.0554	0.0443	0.103
512	0.0778	0.0725	0.1308
1024	0.121	0.1085	0.1543





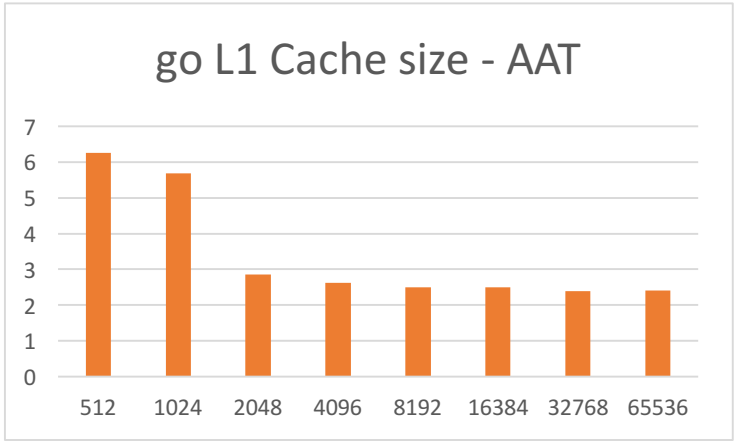
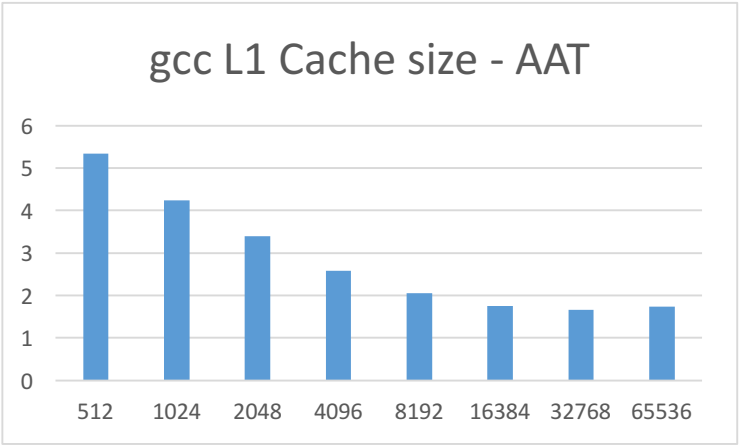
结论：缺失率随块大小的增加先降低后升高，因此可以找到最优的块大小使缺失率最低。

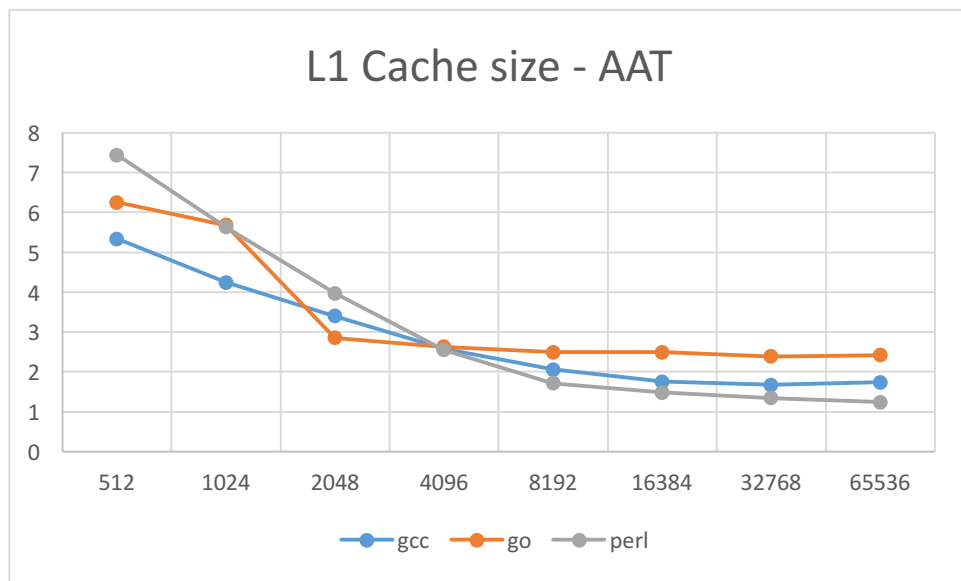
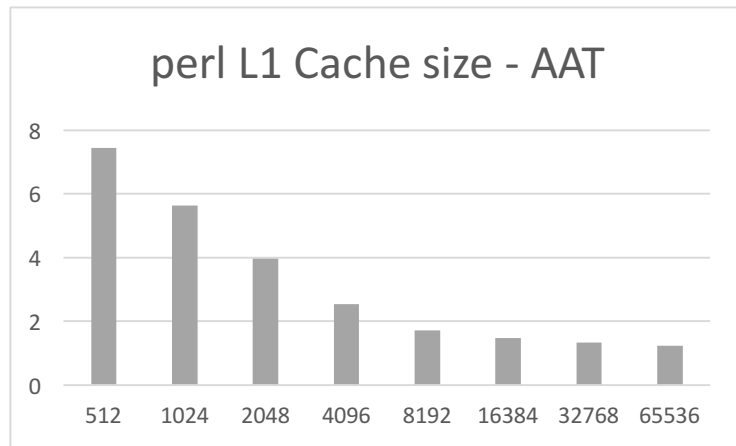
## 二、探索 Cache 设计空间并讨论其性能变化趋势

### 1、L1 Cache size vs. AAT

块大小为 16B，相联度为 1，替换策略为 LRU，写策略为 WBWA 时：

Cache/B	gcc	go	perl
512	5.3338	6.2561	7.4446
1024	4.2452	5.6812	5.6337
2048	3.3969	2.8569	3.9676
4096	2.5795	2.6315	2.5446
8192	2.06	2.492	1.7158
16384	1.7551	2.4956	1.4777
32768	1.6686	2.3839	1.3408
65536	1.7324	2.4173	1.2404





结论：平均防存时间随 Cache 空间的增加而降低，但 Cache 空间达到一定限度时对访存时间的影响将不明显。

## 2、Associativity vs. AAT

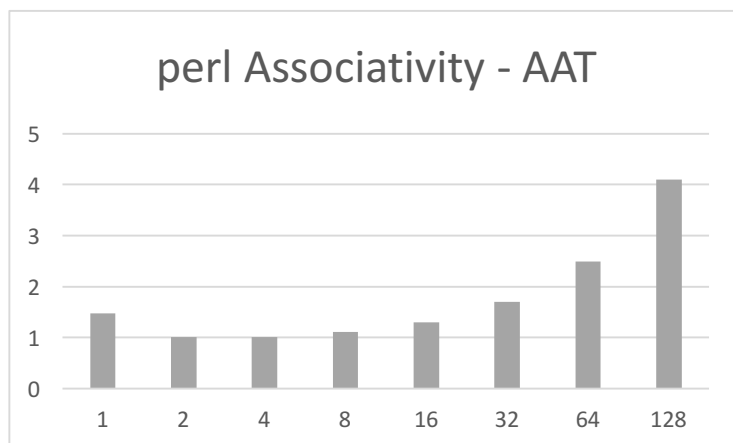
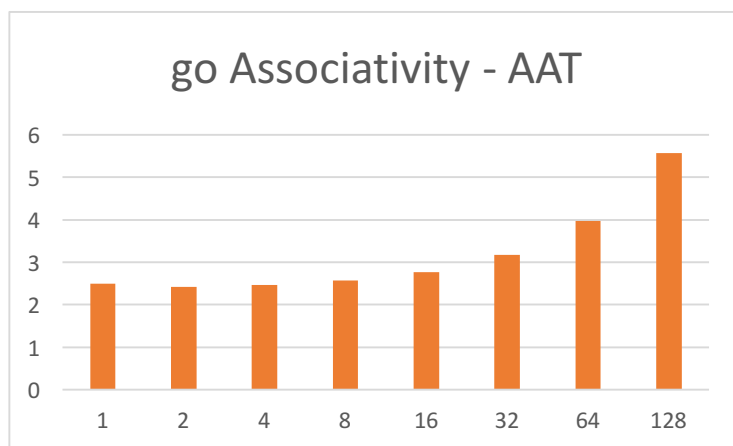
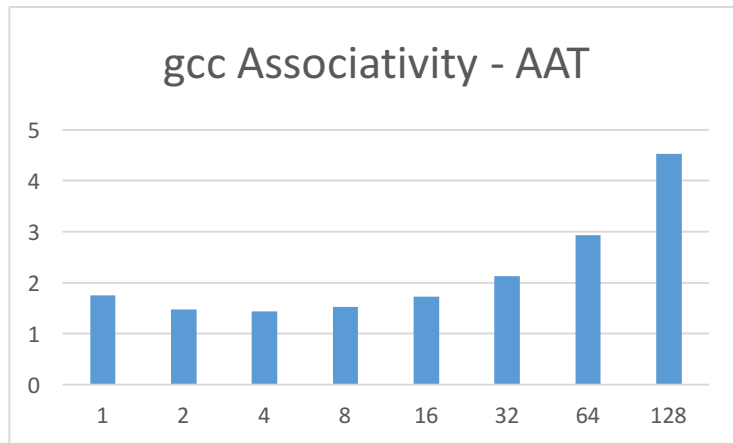
块大小为 16B，Cache 大小为 16384B，替换策略为 LRU，写策略为 WBWA

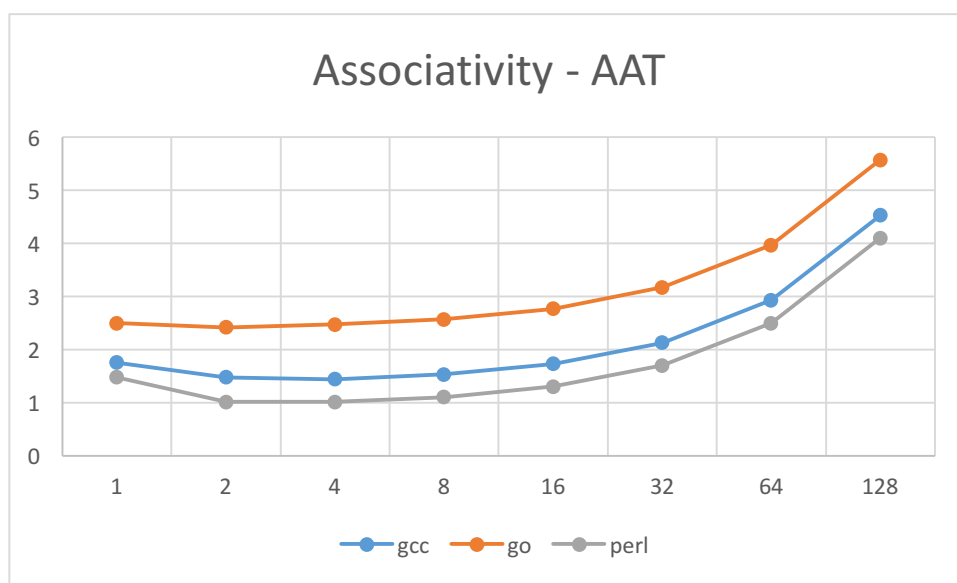
时：

Assoc	gcc	go	perl
1	1.7565	2.4956	1.4777
2	1.48	2.4203	1.012
4	1.4418	2.4697	1.0138
8	1.5312	2.5695	1.1062



16	1.7287	2.7695	1.2997
32	2.1279	3.1693	1.6974
64	2.9277	3.9693	2.4968
128	4.5277	5.5693	4.0968



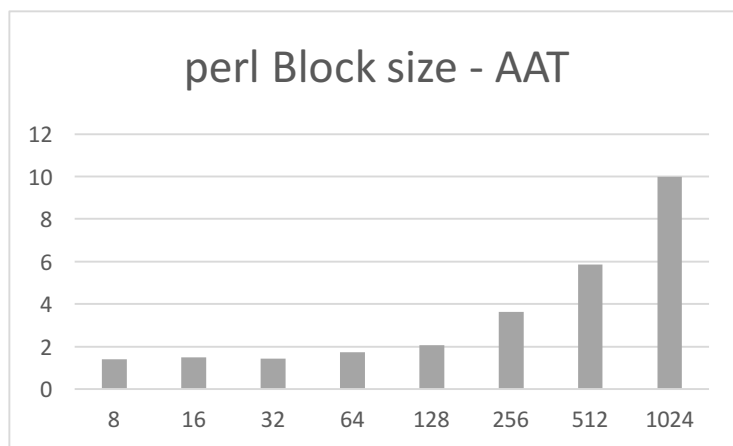
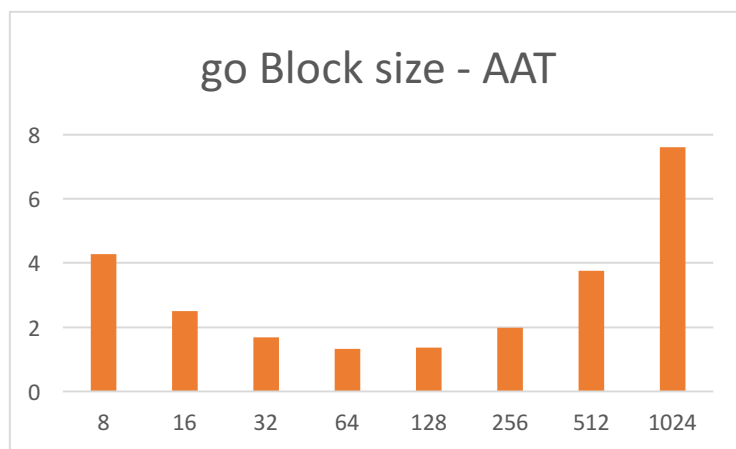
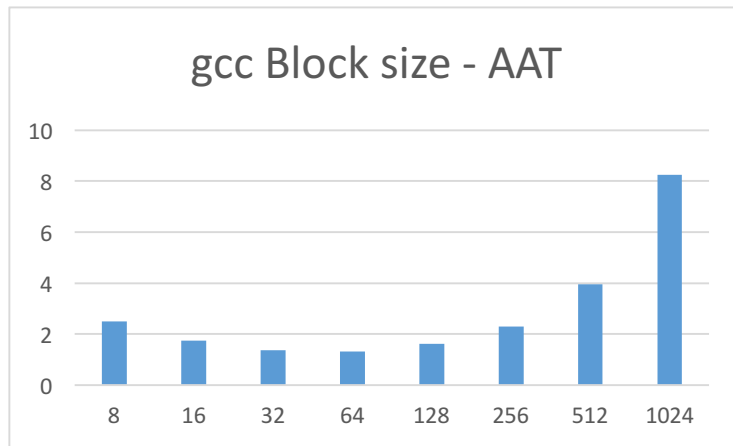


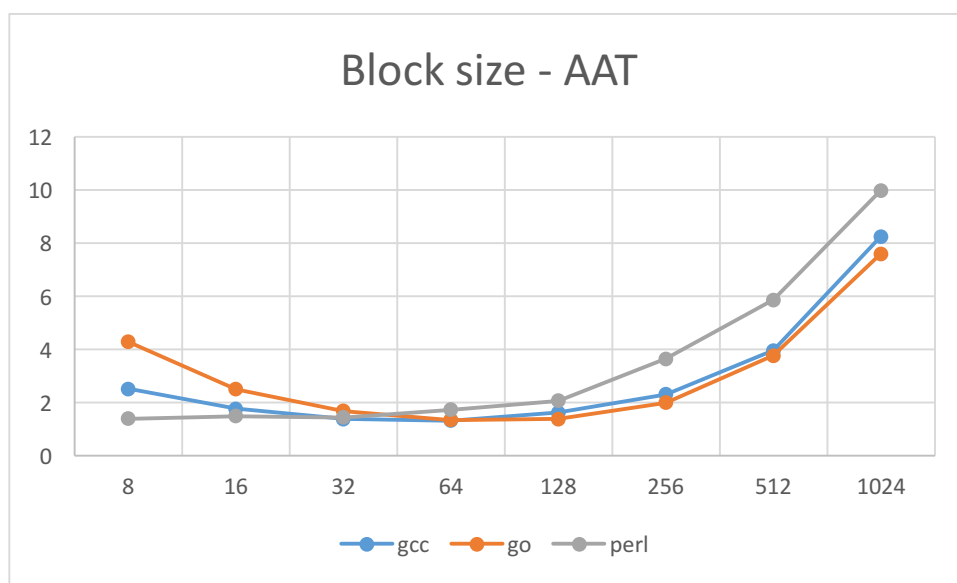
结论：平均防存时间随相联度的增加先降低后升高，因此可以找到最优的相联度使访存时间最短。

### 3、Block size vs. AAT

Cache 大小为 16384B，相联度为 1，替换策略为 LRU，写策略为 WBWA 时：

Block/B	gcc	go	perl
8	2.5054	4.2743	1.3876
16	1.7551	2.4956	1.4777
32	1.371	1.677	1.4367
64	1.3179	1.3353	1.7267
128	1.611	1.3751	2.067
256	2.3038	1.9944	3.638
512	3.9528	3.7642	5.8634
1024	8.2467	7.5936	9.9767





结论：平均防存时间随块大小的增加先降低后升高，因此可以找到最优的块大小使访存时间最短。

### 三、寻找最优的 Cache 存储体系配置方案

由于平均访存时间随 Cache 空间的增加降低，则 Cache 可取上限 512KB；访存时间随相联度的增加先下降后上升，则可以从趋势图中读出最优相联度；访存时间随块大小的增加先下降后上升，则可以从趋势图中读出最优块大小。

综上可得最优结果为：

Trace	Cache size	Assoc	Block size
gcc	512KB	4	64B
go	512KB	2	64B
perl	512KB	2	32B