

天津大学

计算机组成原理实践课程 实验一、单级 Cache 仿真器设计



学 院 智能与计算学部
专 业 计算机科学与技术
学 号 3019244266
姓 名 李润泽

实验在完成 cache 的过程中，对该体系结构进行分析，实验讨论如下：

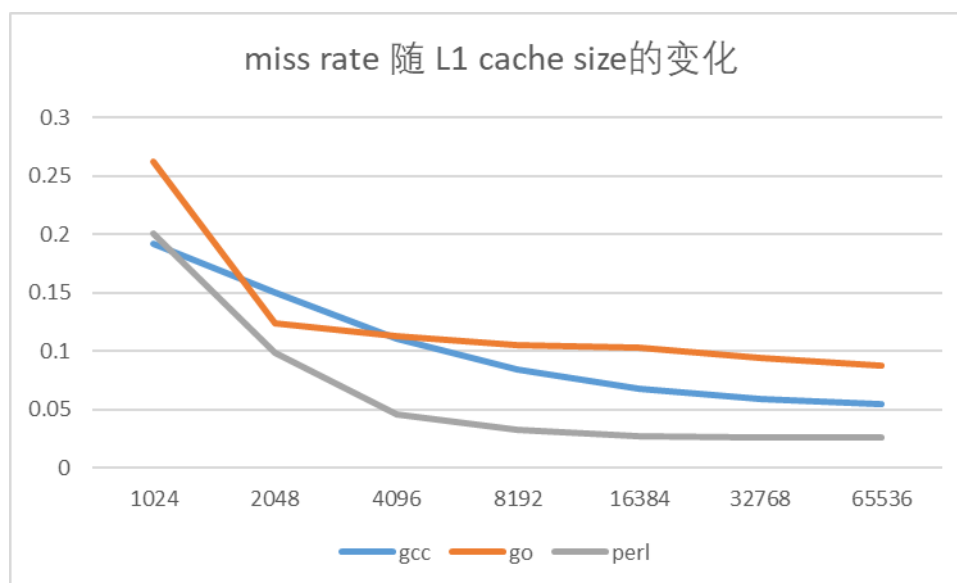
一、分析讨论各种体系结构参数对于 Cache 缺失率的影响

1、L1 Cache size vs. miss rate

根据设定，我们预设 block size = 16B，assoc = 1，替换策略为 LRU，写策略为 WBWA，将 cache size 设定为不同值，获得的 miss rate 值如下表所示：

L1 Cache size	gcc	go	perl
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

miss rate 随 L1 Cache size 的变化如下图所示：



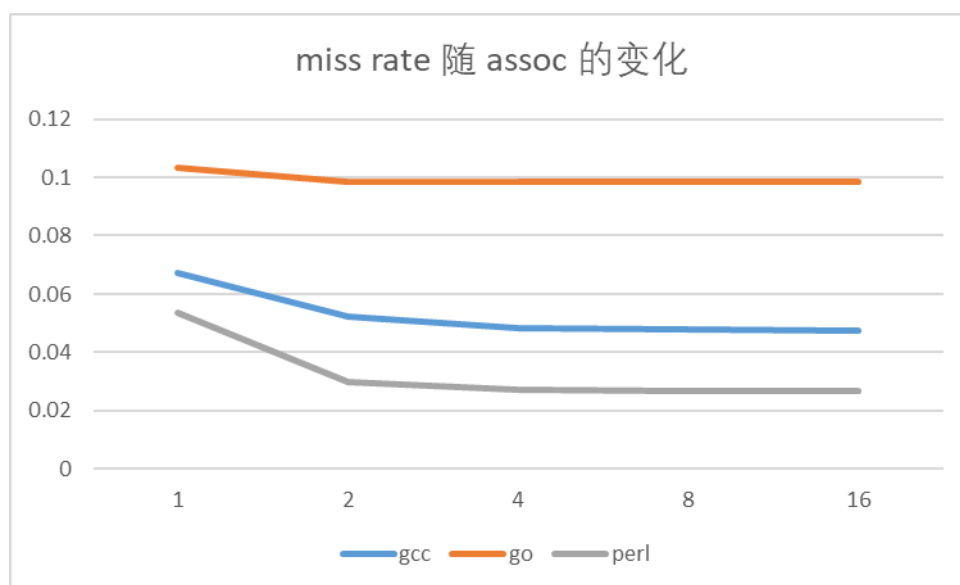
从图像可以看出，缺失率 miss rate 随着 L1 Cache size 的增加而降低，但当 L1 Cache size 过大时，miss rate 降低的就不太明显。

2、Associativity vs. miss rate

根据设定，我们预设 block size = 16B，L1 Cache size = 16384B，替换策略为 LRU，写策略为 WBWA，将 assoc 设定为不同值，获得的 miss rate 值如下表所示：

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

miss rate 随 assoc 的变化如下图所示：



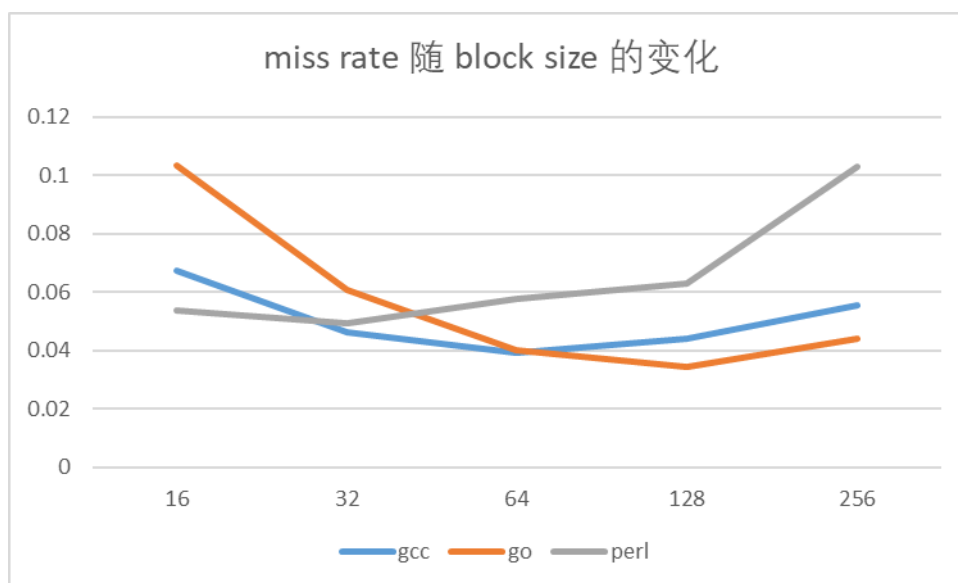
从图像可以看出，虽然缺失率 miss rate 随相联度 assoc 的上升而略微下降，但不明显，而且当 $\text{assoc} > 2$ 时，miss rate 不再变化。

3、Block size vs. miss rate

根据设定，我们预设 L1 Cache size = 16384B， $\text{assoc} = 1$ ，替换策略为 LRU，写策略为 WBWA，将 block size 设定为不同值，获得的 miss rate 值如下表所示：

Block size	gcc	go	perl
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

miss rate 随 block size 的变化如下图所示：



从上图可以看出，缺失率 miss rate 随着块大小 block size 的增加而降低，随后升高，其中，当 block size = 64B 时，gcc 测试的 miss rate 最小；当 block size = 128B 时，go 测试的 miss rate 最小；当 block size = 32B 时，perl 测试的 miss rate 最小。

二、探索 Cache 设计空间 (design space)，讨论其性能变化趋势、

该部分中，实验利用开发的仿真器，探索 Cache 存储体系的设计空间，收集每种配置下的相关仿真效果 (各种性能统计指标和 AAT)。其中，具体讨论如下：

- (1) 采用图形或表格方式，解释随着 Cache 存储体系配置参数的变化，其性能指标 (AAT) 的变化趋势；
- (2) 讨论性能指标的各种变化趋势如何收到各类配置参数变化的影响。

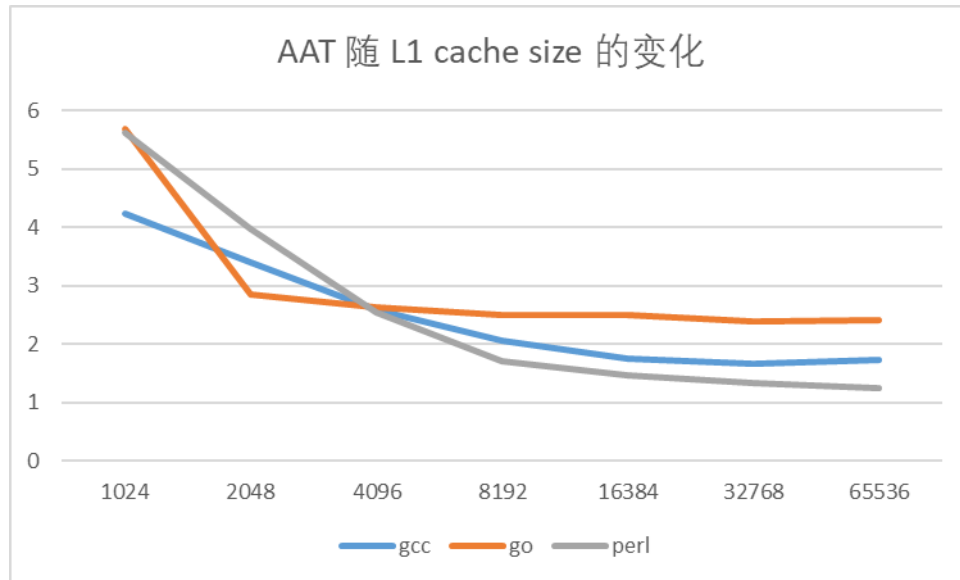
1、L1 Cache size vs. AAT

根据设定，我们预设 block size = 16B，assoc = 1，替换策略为 LRU，写策略为 WBWA，将 cache size 设定为不同值，获得的 AAT 值如下表所示：

L1 Cache size	gcc	go	perl
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

AAT 随 L1 Cache size 的变化如下图所示：



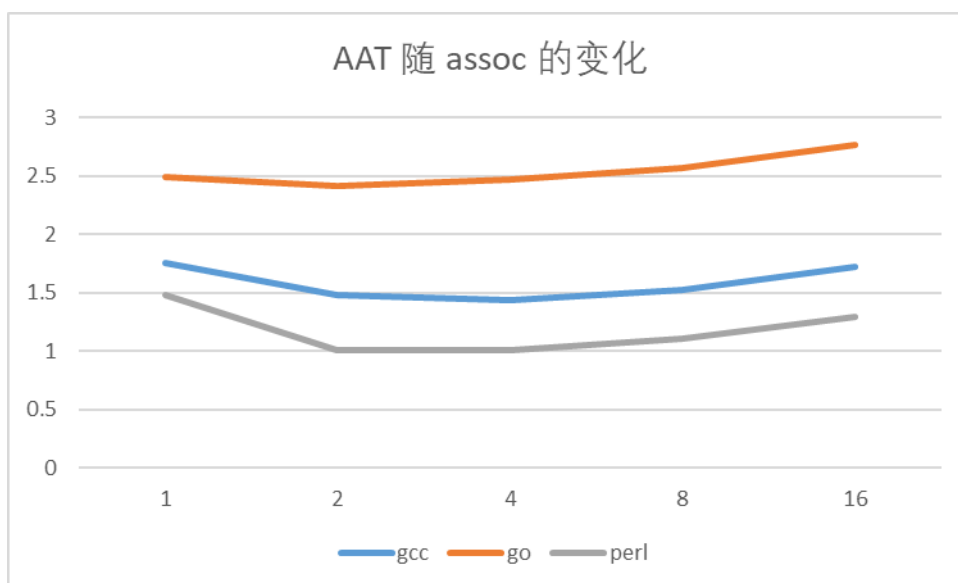
从上图可以看出，平均访存时间 AAT 随着 L1 Cache size 的增加而降低，但当 L1 Cache size > 4096B 时，其对 AAT 的影响就不太明显。

2、Associativity vs. AAT

根据设定，我们预设 block size = 16B，L1 Cache size = 16384B，替换策略为 LRU，写策略为 WBWA，将 assoc 设定为不同值，获得的 AAT 值如下表所示：

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

AAT 随 assoc 的变化如下图所示：



从上图可以看出，平均访存时间 AAT 随着 assoc 的增加而略微上升。

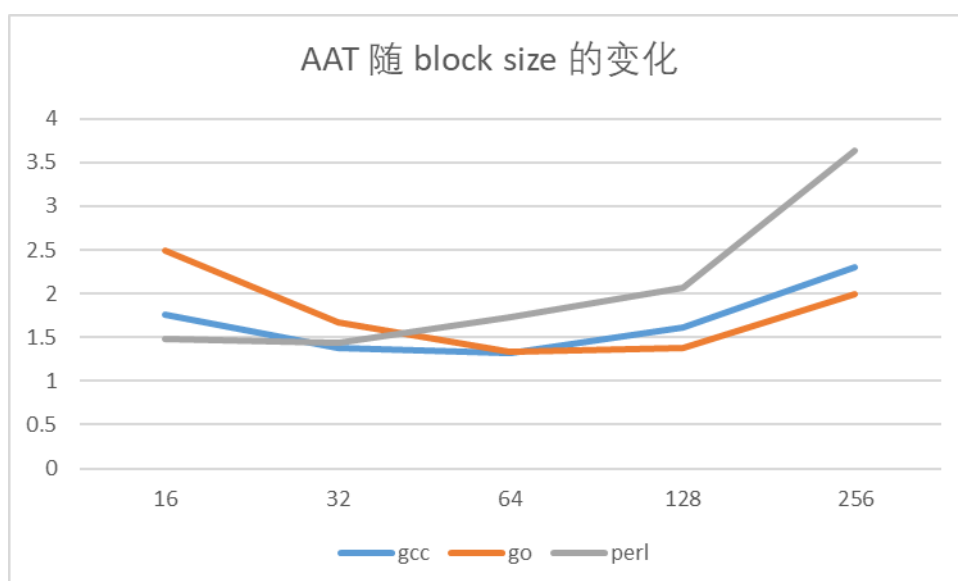
3、Block size vs. AAT

根据设定，我们预设 L1 Cache size = 16384B，assoc = 1，替换策略为 LRU，

写策略为 WBWA，将 block size 设定为不同值，获得的 AAT 值如下表所示：

Block size	gcc	go	perl
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

AAT 随 block size 的变化如下图所示：



从上图可以看出，平均访存时间 AAT 随着 block size 的增加而先降低后增加，其中，当 block size = 64B 时，gcc 测试的 AAT 最小；当 block size = 64B 时，go 测试的 AAT 最小；当 block size = 32B 时，perl 测试的 AAT 最小。

三、寻找最优的 Cache 存储体系配置方案

通过设计空间，实验需要找出针对每种 benchmark trace 的最优 Cache 存储体系配置方案，即在满足面积约束下 ($\text{Area} \leq \text{Area Budget}$)，AAT 最小的配置。

从上文可以看出，平均访存时间 AAT 随 L1 Cache size 的增加而略微降低，故 L1 Cache size 可以取 512KB；平均访存时间 AAT 随着 assoc 或 block size 的增加而先下降后上升，故都可以根据不同 benchmark trace 的情况来选择合适的最优值。

最终可以得出如下结果：

- (1) 对于 gcc 测试，L1 Cache size = 512KB，assoc = 4，block size = 64B；
- (2) 对于 go 测试，L1 Cache size = 512KB，assoc = 2，block size = 64B；
- (3) 对于 perl 测试，L1 Cache size = 512KB，assoc = 2，block size = 32B。