

计算机组织与体系结构实习报告 Lab3.2

姓名：盛朱恒

学号：1600012830

cache管理策略优化（70分）

根据Lab 3.2实习指导的要求，对默认配置下Cache进行优化。并使用附件中所给测试trace，对优化前后的cache进行比较。

1. 请填写以下参数。（10分）

- 默认配置下，32nm工艺节点下，L1 Cache的 Hit Latency 为1.47944ns，约等于3cycle
- 默认配置下，32nm工艺节点下，L2 Cache的 Hit Latency 为1.9206ns，约等于4cycle

2. 默认配置下，运行trace2017中的两个trace，结果如下：（20分）

默认配置：替换算法为LRU，采用写回、写分配，没有旁路以及预取。

局部Miss Rate：访问这一层后Miss的数量/访问这一层的次数

全局Miss Rate：1 - 在该层前已经Hit的次数/总的访存操作次数

01-mcf-gem5-xcg

- 运行trace共1遍，算法中没有随机，因此每次结果都一样。
- L1 Cache：局部Miss Rate = 0.200455，全局Miss Rate = 0.200455
- L2 Cache：局部Miss Rate = 0.424029，全局Miss Rate = 0.096135
- AMAT = 16.381439

02-stream-gem5-xaa

- 运行trace共(1)遍，算法中没有随机，因此每次结果都一样。
- L1 Cache：局部Miss Rate = 0.113404，全局Miss Rate = 0.113404
- L2 Cache：局部Miss Rate = 0.755202，全局Miss Rate = 0.113404
- AMAT = 18.703333

注意：该部分数值应与实际运行结果一致

3. 请填写最终确定的优化方案，并陈述理由。对于涉及到的算法，需要详细描述算法设计和实现思路，并给出优缺点分析。（20分）

旁路部分，采用维护一个表的方法，维护一个大小为 2^X 的表。这个表记录的是地址的第b+1位到第b+X位为某一值（对应 2^X 的表大小）的在缓存中的个数。如果查询个数为0，那么这一地址肯定不在缓存中，否则，可能会在缓存中。理由：这一方法可以有效降低局部Miss Rate。

预取部分，采用预取下若干行的方法，最后采用的是预取下1行。理由，这一方法可以更好的利用局部性，可能会对部分情况下有效。同时发现预取1行有较大提升，后续提升较小，并且trace02有AMAT上升趋势，为了避免过度拟合，所以只预取下一行。

其他替换算法，尝试了两个替换算法：

LRU-K算法：K为设置的值。记录每一行进入cache后被访问的次数，优先留下次数大于或等于K的行。也就是在寻找替换行时，先寻找次数等于1的行，在它们中使用LRU算法，然后次数依次增加，对次数大于等于K的值则在一起使用LRU。根据查询到的资料使用的是LRU-2。

LIRS算法：每一行设置IRR和R值。IRR值为每一行的最近两次访问间隔，（如果只被访问过一次，那么间隔视为无穷大），R值为上次访问到现在的间隔。选择替换行采用先看IRR，选择较大的进行替换，如果IRR相同，选择R较大的替换。

最后决定使用LIRS算法，因为它在相比之下表现稍好一点点。并且分析两种本质上比较相似。LRU-2中，行分为访问过一次的行和访问过两次以上的行，优先替换掉只访问了一次的行。LIRS中的行也可以分为IRR为无穷大的行（只访问过一次的行）和IRR有值的行（访问过两次以上的行），优先替换前者。并且IRR对访问间隔更加细分，所以大部分情况下可能会效果比较好。

4. 优化配置下，运行trace2017中的两个trace，结果如下：（20分）

优化配置：替换算法为LIRS，采用写回、写分配，采用旁路算法，预取算法为预取下一行。

01-mcf-gem5-xcg

- 运行trace共1遍，算法中没有随机，因此每次结果都一样。
- L1 Cache：局部Miss Rate = 0.001567，全局Miss Rate = 0.164558
- L2 Cache：局部Miss Rate = 0.006205，全局Miss Rate = 0.057091
- AMAT = 9.535851
- AMAT降低了41.78%

02-stream-gem5-xaa

- 运行trace共1遍，算法中没有随机，因此每次结果都一样。
- L1 Cache：局部Miss Rate = 0.000247，全局Miss Rate = 0.004395
- L2 Cache：局部Miss Rate = 0.000037，全局Miss Rate = 0.000055
- AMAT = 3.037766
- AMAT降低了83.75%

注意：该部分数值应与实际运行结果一致

cache性能优化测试排名（30分）
