

计算机组织与体系结构实习报告 Lab3.1

学号：1400012783

姓名：王君珊

大班教师：程旭

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（ 1.4794 ） ns，约等于（ 3 ） cycle
- 默认配置下，32nm工艺节点下，L2 Cache的 Hit Latency 为（ 1.9206 ） ns，约等于（ 4 ） cycle

2. 默认配置下，运行两个trace，结果如下：（10分）

- 01-mcf-gem5-xcg.trace:
 - 运行trace共(10)遍
 - L1 Cache：平均 Miss Rate = （ 0.199981 ）
 - L2 Cache：平均 Miss Rate = （ 0.409665 ）
 - AMAT =(6.596237)
- 02-stream-gem5-xaa.trace:
 - 运行trace共(10)遍
 - L1 Cache：平均 Miss Rate = （ 0.113404 ）
 - L2 Cache：平均 Miss Rate = （ 0.999946 ）
 - AMAT =(7.736893)

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。（40分）

- 替换算法（Replacement）
 - LRU：最近最少使用策略。
 - 算法设计实现：为每个line保存一个count标签标记最后一次被使用的时间。当某个line被访问或被替换时其count置0，而同一set其他line的count加1。当需要替换时，遍历set中

的所有line，找到count最大的，即最长时间没有被访问过的line，进行替换。

- 额外存储开销：每个line维护一个count标签，开销为32bit。
- 优缺点：利用了局部性规律，合理，命中率高；但是不能保证被替换的块不会再被使用，且系统实现复杂。

- FIFO：先进先出策略。

- 算法设计实现：为每个line保存一个come_time标签记录数据进入的时间。当新数据进入该line时其come_time置0，每一个时钟周期所有line的come_time加1。当需要替换时，遍历set中的所有line，找到come_time最大的，即最早进入的line，进行替换。
- 额外存储开销：每个line维护一个come_time标签，开销为32bit。
- 优缺点：易于实现；但不符合局部性规律，即被频繁访问的数据块会被替换。

- LFU：最不经常使用策略。

- 算法设计实现：为每个line保存一个hit_times标签记录数据进入的时间。当新数据进入该line时其hit_times置0，每次访问该line的数据时其hit_times加1。当需要替换时，遍历set中的所有line，找到hit_times最小的，即被访问次数最少的，进行替换。
- 额外存储开销：每个line维护一个hit_times标签，开销为32bit。
- 优缺点：被频繁访问的数据块会留在cache中；但新进入的数据块容易被替换出去。

- RANDOM：随机策略

- 算法设计实现：每次随机选择set中的某一line进行替换。
- 额外存储开销：无。
- 优缺点：方法简单，易于实现；但是命中率低。

- 数据预取 (Prefetch)

- 算法设计实现：prefetch策略是在每次从下一级缓存获取数据（即该级cache发生miss）时，预先从下一级获取紧接着的三个block。首先判断需要prefetch的数据是否已经在cache中，如果不在则从下一级获取。需要更新数据块、相关替换策略标记等信息。
- 额外存储开销：无。
- 优缺点：利用了空间局部性的原理，预先获取更多数据，减少多次访问下级存储的开销。

- 旁路策略 (Bypass)

- 算法设计实现：bypass策略是不再从当前cache获取数据，而是直接从下一级cache获取。为每个set保存一个miss和total分别表示对于该set的访问miss的次数和总次数，当访问总次数大于一个阈值且miss的比例也大于一个阈值时，发生bypass，直接调用下一级cache的HandleRequest函数。
- 额外存储开销：每个set维护一个miss和total标签，开销一共64bit。
- 优缺点：对于miss比例高的set，其数据不在cache中且不会再被访问，此时不需要存入cache中，通过旁路策略可以减少多余的访存操作。

L1 Cache：使用LFU替换策略，使用数据预取prefetch，使用旁路策略bypass。

L2 Cache：使用LFU替换策略，使用旁路策略bypass。

4. 优化配置下，运行两个trace，结果如下：（10分）

◦ 01-mcf-gem5-xcg.trace:

- 运行trace共(10)遍
- L1 Cache: 平均 Miss Rate = (0.159360)
- L2 Cache: 平均 Miss Rate = (0.042432)
- AMAT =(2.635701)

◦ 02-stream-gem5-xaa.trace:

- 运行trace共(10)遍
- L1 Cache: 平均 Miss Rate = (0.029613)
- L2 Cache: 平均 Miss Rate = (0.218976)
- AMAT =(1.972319)