

Lab4 实验报告

181220063, 薛正海, xuezh@smail.nju.edu.cn

2019 年 12 月 8 日

1 代码实现

我在 `common.h` 中加入了一些宏定义, 主要代码在 `cache.c` 和 `utils.c` 中, 能通过随机测试. 一个在我看来 non-trivial 的点是计算块内地址时都需按照四字节对齐. 另外, 访问 cache 时程序记录了命中与缺失的次数及总的 cycle 数. 统计时利用 Python 批量运行代码, 分析数据并作图. 为解决内存序列较长, 运行速度较慢的问题, 我在 `run_cycles_stat.py` 中引入了 multiprocessing 模块, 充分利用 cpu 资源.

2 实验数据收集

2.1 命中率

cache 命中率通过计算命中次数与 cache 总访问次数之比得到. 实验分别考察了命中率与 cache 大小和 associativity 大小的关系. 与 cache 大小的关系如图1所示.

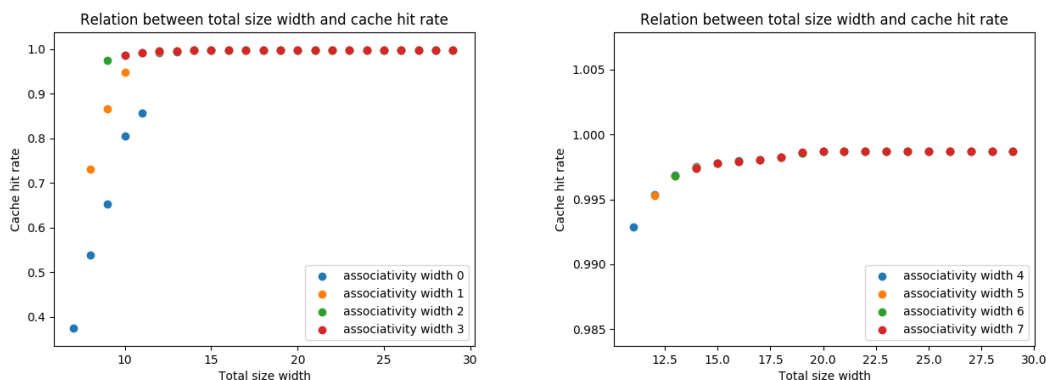


图 1: 命中率与 cache 大小的关系

命中率与 associativity 的关系如图2所示. 可以看出, 当 cache 较小时 cache 的大小与命中率呈近似二次函数的关系, 且 associativity 越大, 命中率越高. 当 cache 较大时 cache 命中率本身较高, 与 cache 大小及 associativity 无明显联系.

2.2 cache 的复杂性

cache 的复杂度不可用程序运行时间来衡量. 这是由于在 cache 模拟器中循环遍历一个 cache 组时, 真正的 cpu cache 会加速这一循环过程, 而真正的 cache 在循环遍历时并没有这一

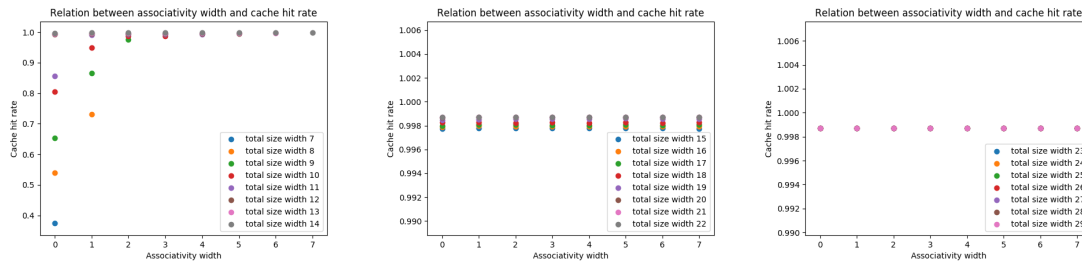


图 2: 命中率与 associativity 的关系

加速效果. 所以 cache 的复杂度应通过平均访问周期数来衡量. 实验分别考察了复杂度与 cache 大小和 associativity 大小的关系. 与 cache 大小的关系如图3所示. 可见在不同 associativity

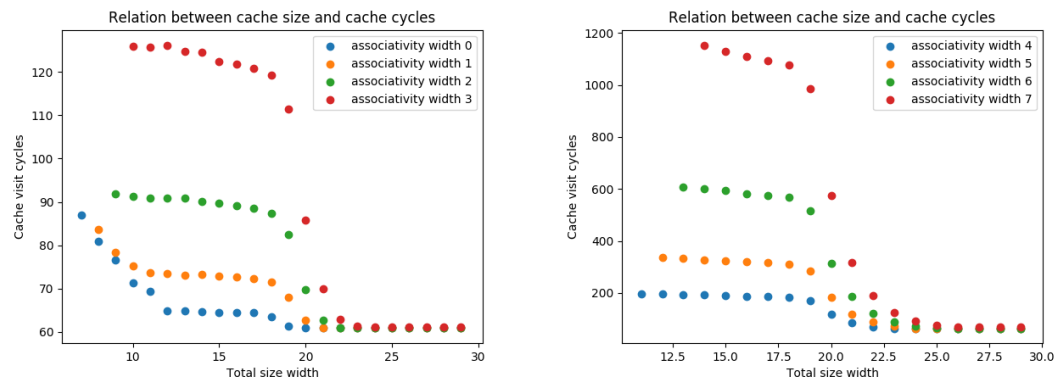


图 3: cache 复杂度与 cache 大小的关系

下,cache 大小都存在一个临界值, 超过这个临界值后复杂性会急剧下降若干倍, 这可能与 cache 的替换有关: 超过临界值后 cache 的替换行为会显著减少.cache 复杂度与 associativity 的关系如图4所示. 可见不同 cache 大小下,associativity 与 cache 复杂性都存在近似二次函数的关系

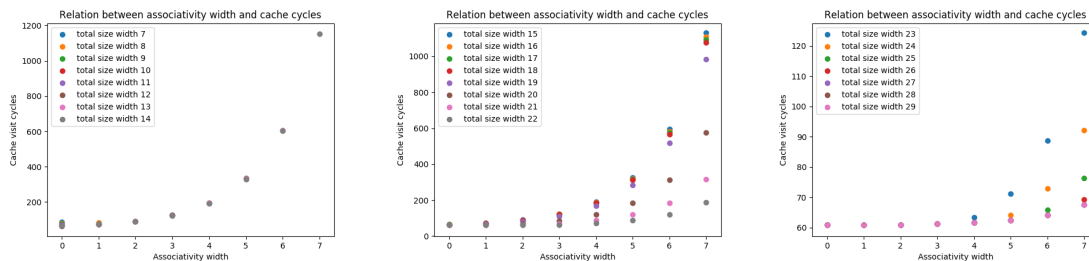


图 4: cache 复杂度与 associativity 的关系

系:associativity 越大,cache 复杂性越高. 考虑到访问 cache 时需要遍历一个组中的所有行, 这点是易见的.

3 性能建模与 cache 设计

3.1 性能分析

由实验数据可知,cache 大小越大,命中率越高,复杂性也越低. 因此 cache 应尽可能大. 但考虑到某些地方存在性能跃迁, 且此后再增大 cache 对性能影响不大,cache 大小应尽可能选在跃迁附近. 综合命中率与复杂性两者, 可初步确定 cache 的数据总宽度在 19 ~ 23 之间, 即大小在 512KiB ~ 8MiB 之间.

而对 associativity 而言, 过大的 associativity 会导致 cache 复杂度提升, 但提升的幅度不一:cache 大小越大, 相对提升的幅度越小. 因此越大的 cache 应选择更大的 associativity. 对于较小的 cache, 过小的 associativity 还会导致命中率降低, 因此选择时需要一些 trade-off.

3.2 联系实际的 cache 设计

使用 `cat /sys/devices/system/cpu/cpu0/cache/index0/size` 来查看系统中各级 cache 的大小. 我的系统中 cache 大小数据为:

L1d cache	L1i cache	L2 cache	L3 cache
32KiB	32KiB	256KiB	8192KiB

其中 L1 cache 分为指令 cache 和数据 cache, 在 cache 模拟中未被体现, 故暂不考虑. 对 L3 cache 而言, 其容量大,associativity 较小时其对 cache 复杂度影响不大. 如图4所示,associativity 在 16 以下时 cache 复杂度基本不变, 继续增大时复杂度增大较多, 故选择 associativity 为 16.

而对 L2 cache 而言, 其容量较小,associativity 对复杂度影响较大, 对命中率影响却不大, 因此应选择较小的 associativity. 参考图4, 选择 associativity 为 4.

3.3 结论

针对 microbench-test 的 workload, 我认为最佳的 L2 cache 设计为 256KiB 的 cache 大小,4 路组相连; 最佳的 L3 cache 设计为 8192 KiB 的 cache 大小,16 路组相连.

以上就是我本次实验报告的全部内容, 感谢您的阅读!