

Cache 系统组织与设计实验报告

计 76 陈之杨 2017011377

2020.4

1 使用方法

`cache_sim.cpp` 为源代码，使用 `g++ -o cache_sim cache_sim.cpp -O2` 命令编译。运行 `cache_sim` 进行 cache 模拟，参数如下所示：

- `-block [blocksize]`，设置缓存块大小为 `blocksize`。
- `-alloback/allothro/aroback/arothro`，设置写策略为写分配-写回/写分配-写直达/写绕过-写回/写绕过-写直达。
- `-full/direct/4way/8way`，设置映射规则为全关联/直接映射/4 路组关联/8 路组关联。
- `-lru/random/tree`，设置替换策略为 LRU/随机替换/二叉树替换。
- `-log [filename]`，输出日志到 `filename` 文件中。

当某项参数缺省时，采用默认 cache 布局（块大小 8B，8 路组关联，LRU 替换策略，写分配，写回）。提交的 `astar.log` 等文件为默认布局下，给定 `trace` 文件的访问日志。程序默认从标准输入读入，如果要从指定 `trace` 文件中读入，需要重定向输入，如 `./cache_sim < astar.trace`。

程序运行完毕后，会在标准输出中打印 cache 命中率。

2 实现细节

程序中定义了 `Cache` 类作为访问的接口，内部定义了一个 `Group` 类数组，描述 cache 中的每个组（将直接映射视作 1 路组关联，全关联将整个 cache 视为一个组）。访问某一内存地址时，`Cache` 类负责提取索引位，找到对应的组，然后将具体的访问交给 `Group` 类执行。

每个 `Group` 类中，用一个二维 `char` 数组 `metaData` 维护每个块的元数据，再定义一个 `Replace` 类执行替换策略。

Replace 类是一个抽象类，只定义了 insert,access,replace 三个接口的形式，对于不同的替换策略，分别定义一个类继承 Replace 实现接口。例如，LeastRecentlyUsed 类中定义了 char 数组 stack 实现 LRU 的堆栈法，BinaryTree 类中定义了 char 数组 tree 维护二叉树信息，其中二叉树顶点用完全二叉树的方式组织，即顶点 p 的左右儿子分别为 2p 和 2p+1，这样整个二叉树可以用组大小个连续空间实现。

所有 cache 系统中需要维护的信息都用 char 数组存储，例如 8 路组关联的 LRU 策略需要 24 位，使用 3 个 char 存储，8 路组关联的二叉树策略需要 8 位，使用 1 个 char 存储（由于二叉树的根从 1 开始，需要额外的 1 位）。8 路组关联 8B 块的元信息使用 7 个 char 存储。为了方便从 char 数组中读取或写入连续一段二进制位，实现了 editBits 和 readBits 两个函数统一操作。

3 实验结果

由于所有的参数设置都需要测试 4 个重点 trace 的结果，以下用 a/b/c/d 的形式表示 astar,bzip2,mcf,perlbench 四个重点 trace 的命中率分别为 a%,b%,c%,d%（保留四位有效数字，加粗为最优命中率）。

3.1 组织方式

组织 \ 块大小	8B	32B	64B
直接映射	76.60/97.94/95.06/96.33	90.16/98.67/97.80/97.69	94.73/98.41/98.54/98.11
4 路组关联	76.72/98.78/95.42/97.93	90.31/99.69/98.18/98.86	94.99/99.85/98.92/99.15
8 路组关联	76.72/98.78/95.42/98.21	90.37/99.69/98.18/99.18	95.00/99.85/98.92/99.38
全关联	76.74/98.78/95.42/98.24	90.41/99.69/98.18/99.34	95.03/99.85/98.92/99.61

3.2 替换策略

替换策略	命中率
LRU	76.72/ 98.78/95.42 /98.21
随机	76.77 /98.78/95.40/ 98.22
二叉树	76.71/98.78/95.42/98.22

3.3 写策略

写策略	写回	写直达
写分配	76.72/98.78/95.42/98.21	76.72/98.78/95.42/98.21
写绕过	65.50/91.33/88.85/95.34	65.50/91.33/88.85/95.34

