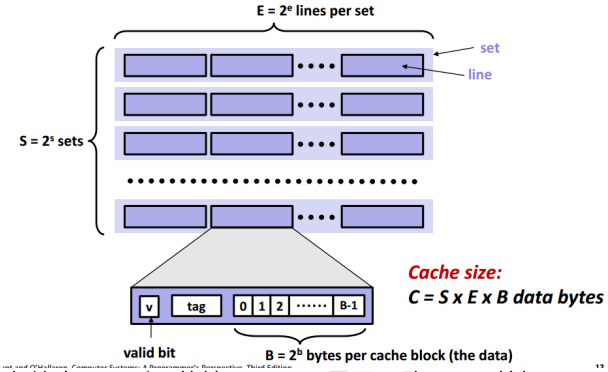
15213-ICS-Cache

2017年8月24日 21:50

Valgrind 命令：
Valgrind是一个调试和剖析的程序工具集。
valgrind --log-fd=1 --tool=lackey -v --trace-mem=yes ls –l
打印出ls -l 命令访问的所有内存

表示这个真不太会，所以我又偷懒去看大神是怎么做的了：https://github.com/LewisVo/Cache-Lab

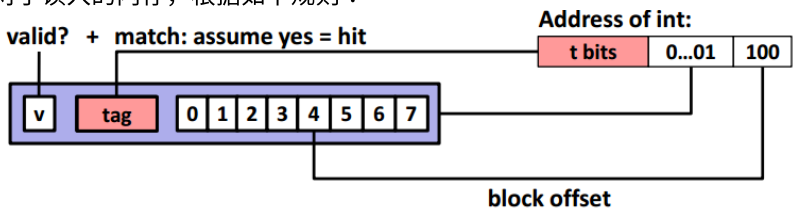
Part A
要模拟cache，我们得先有一个cache:



在给定s，E, 和b的情况下，只需要一路malloc就好了。

```
newCache.sets = (struct cacheSet*) malloc(sizeof(struct cacheSet) * numberOfSets);
set.lines = (struct line*) malloc(sizeof(struct line) * numberOfLines);
```

然后我们打开trace文件，依次读入内存address。
对于读入的内存，根据如下规则：



计算出其对应的tag，和在cache中的set位置：

```
int tagSize = (64 - (exampleParameter.s + exampleParameter.b));
memoryAddress inputTag = address >> (exampleParameter.s + exampleParameter.b);
unsigned long long temp = address << (tagSize);
unsigned long long indexOfSet = temp >> (tagSize + exampleParameter.b);
```

如果valid == 1并且tag匹配，记一次hit, 否则记一次miss。
然后查看cahceSet是否已经满了，如果已经满了，用LRU算法找出使用频率最小的line，覆盖这个位置，evicts++。否则查找空的line，然后把这个内存信息写入这个line。

最后，因为里面用了大量的malloc，所以在最后一定要全部free。

结果：

```
hao@hao:~/Documents/CMU/cmu-15213/Cache Lab/cachelab-handout$ ./test-csim
                        Your simulator     Reference simulator
Points (s,E,b)    Hits  Misses  Evicts    Hits  Misses  Evicts
     2 (1,1,1)      21       8       6       9       8       6  traces/yi2.trace
     2 (4,2,4)       6       5       2       4       5       2  traces/yi.trace
     2 (2,1,4)       8       3       1       2       3       1  traces/dave.trace
     2 (2,1,3)     251      71      67     167      71      67  traces/trans.trace
     0 (2,2,3)     281      41      33     201      37      29  traces/trans.trace
     2 (2,4,3)     296      26      10     212      26      10  traces/trans.trace
     2 (5,1,5)     315       7       0     231       7       0  traces/trans.trace
     4 (5,1,5)  303163   21775   21743  265189   21775   21743  traces/long.trace
    16

TEST_CSIM_RESULTS=16
```

Part B

这里说cache有32个sets，每个set有一个line，每个line的block size是8 bytes。但是根据如下显示，我不太理解为什么block size是8。

```
hao@hao:~/Documents/CMU/cmu-15213/Cache Lab/cachelab-handout$ ./test-trans -M 32 -N 32

Function 0 (2 total)
Step 1: Validating and generating memory traces
Validation error at function 0! Run ./tracegen -M 32 -N 32 -F 0 for details.
Skipping performance evaluation for this function.

Function 1 (2 total)
Step 1: Validating and generating memory traces
Step 2: Evaluating performance (s=5, E=1, b=5)
func 1 (Simple row-wise scan transpose): hits:870, misses:1183, evictions:1151

Summary for official submission (func 0): correctness=0 misses=2147483647

TEST_TRANS_RESULTS=0:2147483647
hao@hao:~/Documents/CMU/cmu-15213/Cache Lab/cachelab-handout$
```

后面先mark一下，后面再写。
先上结果：

```
hao@hao:~/Documents/CMU/cmu-15213/Cache Lab/cachelab-handout$ ./test-trans -M 32 -N 32

Function 0 (2 total)
Step 1: Validating and generating memory traces
Step 2: Evaluating performance (s=5, E=1, b=5)
func 0 (Transpose submission): hits:1766, misses:289, evictions:257

Function 1 (2 total)
Step 1: Validating and generating memory traces
Step 2: Evaluating performance (s=5, E=1, b=5)
func 1 (Simple row-wise scan transpose): hits:870, misses:1183, evictions:1151

Summary for official submission (func 0): correctness=1 misses=289

TEST_TRANS_RESULTS=1:289
hao@hao:~/Documents/CMU/cmu-15213/Cache Lab/cachelab-handout$ ./test-trans -M 64 -N 64

Function 0 (2 total)
Step 1: Validating and generating memory traces
Step 2: Evaluating performance (s=5, E=1, b=5)
func 0 (Transpose submission): hits:9026, misses:1221, evictions:1189

Function 1 (2 total)
Step 1: Validating and generating memory traces
Step 2: Evaluating performance (s=5, E=1, b=5)
func 1 (Simple row-wise scan transpose): hits:3474, misses:4723, evictions:4691

Summary for official submission (func 0): correctness=1 misses=1221

TEST_TRANS_RESULTS=1:1221
hao@hao:~/Documents/CMU/cmu-15213/Cache Lab/cachelab-handout$ ./test-trans -M 61 -N 67

Function 0 (2 total)
Step 1: Validating and generating memory traces
Step 2: Evaluating performance (s=5, E=1, b=5)
func 0 (Transpose submission): hits:6279, misses:1902, evictions:1870

Function 1 (2 total)
Step 1: Validating and generating memory traces
Step 2: Evaluating performance (s=5, E=1, b=5)
func 1 (Simple row-wise scan transpose): hits:3756, misses:4423, evictions:4391

Summary for official submission (func 0): correctness=1 misses=1902

TEST_TRANS_RESULTS=1:1902
```