

Cachegrind

What is cachegrind?

- A cache profiler
- Simulates how a program interacts with a machine's cache hierarchy
- Branch predictor (Optional)
- Performs detailed simulation of the I_1 , D_1 and L_2 caches in CPU
 - I_1 : First level instruction cache
 - D_1 : First level data cache
 - L_2 : Second level unified cache

What is cachegrind?

- Some modern machines have three or four levels of cache
- For these machines Cachegrind simulates the first-level and last-level caches
 - Last-level cache has the most influence on runtime, as it masks accesses to main memory
 - L1 caches often have low associativity, so simulating them can detect cases where the code interacts badly with this cache

What is cachegrind?

- Accurately pinpoint the sources of cache misses in a code
- For each line of source code, with per-function, per-module and whole-program summaries it identifies
 - The number of cache misses
 - Memory references
 - Instructions executed

Cachegrind statistics?

Cachegrind gathers the following statistics (abbreviations used for each statistic is given in parentheses):

- I cache reads (Ir, which equals the number of instructions executed), I1 cache read misses (I1mr) and LL cache instruction read misses (ILmr)
- D cache reads (Dr, which equals the number of memory reads), D1 cache read misses (D1mr), and LL cache data read misses (DLmr)
- D cache writes (Dw, which equals the number of memory writes), D1 cache write misses (D1mw), and LL cache data write misses (Dlmw)
- Conditional branches executed (Bc) and conditional branches mispredicted (Bcm)
- Indirect branches executed (Bi) and indirect branches mispredicted (Bim)

Example

```
#include<stdio.h>
#include<stdlib.h>

int main(){
    int i, *a, *b;
    a = (int *)calloc(100, sizeof(int));
    b = (int *)calloc(100, sizeof(int));
    for (i = 0; i < 100; i++)
        a[i] = i;
    for (i = 0; i < 100; i++)
        b[i] = a[i];
    for (i = 0; i < 100; i++)
        printf("%d ", b[i]);
    free(a);
    free(b);
}
```

```
valgrind --tool=cachegrind prog
==16201== Cachegrind, a cache and branch-prediction profiler
==16201== Copyright (C) 2002-2011, and GNU GPL'd, by Nicholas Nethercote et al.
==16201== Using Valgrind-3.7.0 and LibVEX; rerun with -h for copyright info
==16201== Command: prog
==16201==
--16201-- warning: L3 cache found, using its data for the LL simulation.
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34
35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51
52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68
69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85
86 87 88 89 90 91 92 93 94 95 96 97 98 99 ==16201==
==16201== I refs:    235,189
==16201== I1 misses:    954
==16201== LLi misses:    936
==16201== I1 miss rate:  0.40%
==16201== LLi miss rate: 0.39%
==16201==
==16201== D refs:    81,463 (58,939 rd + 22,524 wr)
==16201== D1 misses:    3,034 ( 2,447 rd +   587 wr)
==16201== LLd misses:    2,470 ( 1,950 rd +   520 wr)
==16201== D1 miss rate:   3.7% (  4.1% +  2.6% )
==16201== LLd miss rate:  3.0% (  3.3% +  2.3% )
==16201==
==16201== LL refs:     3,988 ( 3,401 rd +   587 wr)
==16201== LL misses:    3,406 ( 2,886 rd +   520 wr)
==16201== LL miss rate:   1.0% (  0.9% +  2.3% )
```

Example

```
valgrind --tool=cachegrind --branch-sim=yes prog
==17165== Cachegrind, a cache and branch-prediction profiler
==17165== Copyright (C) 2002-2011, and GNU GPL'd, by Nicholas Nethercote et al.
==17165== Using Valgrind-3.7.0 and LibVEX; rerun with -h for copyright info
==17165== Command: prog
==17165==
--17165-- warning: L3 cache found, using its data for the LL simulation.
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47
48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91
92 93 94 95 96 97 98 99 ==17165==
==17165== I refs:    235,188
==17165== I1 misses:    953
==17165== LLI misses:    935
==17165== I1 miss rate:  0.40%
==17165== LLI miss rate: 0.39%
==17165==
==17165== D refs:    81,463 (58,939 rd + 22,524 wr)
==17165== D1 misses:  3,034 ( 2,447 rd +  587 wr)
==17165== LLD misses:  2,470 ( 1,950 rd +  520 wr)
==17165== D1 miss rate:  3.7% ( 4.1% + 2.6% )
==17165== LLD miss rate:  3.0% ( 3.3% + 2.3% )
==17165==
==17165== LL refs:    3,987 ( 3,400 rd +  587 wr)
==17165== LL misses:  3,405 ( 2,885 rd +  520 wr)
==17165== LL miss rate:  1.0% ( 0.9% + 2.3% )
==17165==
==17165== Branches:    43,122 (41,990 cond + 1,132 ind)
==17165== Mispredicts:  4,652 ( 4,552 cond +  100 ind)
==17165== Mispred rate: 10.7% ( 10.8% + 8.8% )
```