

CSL211-Cache-Assignment-FAQs

From IIT ROPAR's Wiki

Contents

- 1 Assignment Description
- 2 How to compile/use/configure Dinero IV
- 3 Writing matrix multiplication with instrumentation
- 4 Command to run Dinero for different configuration
- 5 Can I automate running Dinero in different configurations

Assignment Description

Objective: Objective of the assignment is (a) to understand cache hit/miss statistics of data cache in various configuration, (b) to experiment with cache specific code optimization.

Procedure: Use Dinero IV cache simulator (<http://pages.cs.wisc.edu/~markhill/DineroIV/>) to get the cache statistics.

Step 1: Write the code of matrix multiplication with variable array size, with instrumentation to get the memory trace. For example, for any array read add "printf("0 %x", &array[i][j])", for any array write operation add "printf("1 %x", &array[i][j]);" With this instrumentation, after execution of matrix multiplication, you will get a memory access trace.

Step 2: Run the Dinero IV simulator with the above input trace. Get the data miss statistics of cache hits and misses. Find all compulsory, capacity, conflict misses for an application.

Use the following configurations

- Cache size: 16kb, 32kb
- Line/Block size: 32byte, 64 byte
- Associativity: 1, 2, 4
- Replacement policy: LRU

Submission 1:

- Use matrix size as, 200x200 and 500x500.
- Get the results for all (2x2x3x2) cases and give your analysis in form of a report.

Submission 2: Optimize matrix multiplication algorithm to improve the results. You can use google but site the reference in your report. Your code optimization may be cache size specific. Repeat the cache analysis done for submission 1.

This is not a group assignment. Upload your submission as single tar file: assignment5_<entry no>.tgz

How to compile/use/configure Dinero IV

Compiling and building Dinero:

```
tar -xvzf d4-7.tar.gz
cd d4-7
./configure
make
```

Writing matrix multiplication with instrumentation

Define SIZE in the beginning of matrix multiplication code, if it is not defined.

```
#ifndef SIZE
#define SIZE 10
#endif
```

Print the memory trace,

```
sum=0;
for(k=0; k < SIZE; k++) {
    sum += A[i][k] * B[k][j];
    printf("0 %p\n", &A[i][k]);
    printf("0 %p\n", &B[k][j]);
}
C[i][j] = sum;
printf("1 %p\n", &C[i][j]);
```

Compiling step: Size can be given as compile time variable using the following command.

```
gcc -DSIZE <value> matrix.c
```

Command to run Dinero for different configuration

For 16KB cache, with 32 byte block size, associativity 2, following command will generate statistics:

```
./dineroIV -l1-dsize 16k -l1-dbsize 32 -l1-dassoc 2 -l1-dccc -informat d < mem.trace
```

Can I automate running Dinero in different configurations

Yes, you may write a shell script (https://en.wikipedia.org/wiki/Shell_script) to automate.

Following may be pseudo code of the script. YOU NEED TO CHANGE THE SYNTAX.

```
for SIZE in {200, 500} {
    gcc -DSIZE $SIZE mm.c -o mm
    ./mm > mm_$SIZE.trace
    for DSIZE in {16k, 32k} {
        for BSIZE in {32, 64} {
            for ASSOC in {1, 2, 4} {
                ./dineroIV -l1-dsize $DSIZE -l1-dbsize $BSIZE -l1-dassoc $ASSOC -l1-dccc
                    -informat d < mm_$SIZE.trace > stat_$SIZE_$DSIZE_$BSIZE_$ASSOC.txt
            }
        }
    }
}
```

Retrieved from "http://10.1.0.78/mediawiki/index.php?title=CSL211-Cache-Assignment-FAQs&oldid=4300"

- This page was last modified on 11 November 2016, at 09:21.
- This page has been accessed 271 times.