DineroIV Multicore Cache Simulator

Presented by
Brandon Potter
Computer Systems Research Lab (CSRL)
02/02/12

History

- Developed by Jan Edler (NEC) and Mark Hill (Madison-Wisconsin)
- Written September 1997 – February 1998
- Written in C
- Free for academic use
- Vanilla version - uniprocessor cache simulator
- Trace driven
- Often cited in academic research papers

CSRL Work

- Fixed a few bugs with 64 bit addresses; original worked with 32 bit addresses
- Extended DineroIV to multi-core cache simulations
  - Implemented coherency directory (double hash table containing MESI states)
  - Reworked the interconnect between cores
- DineroIV generally used as a background tool to support our other projects

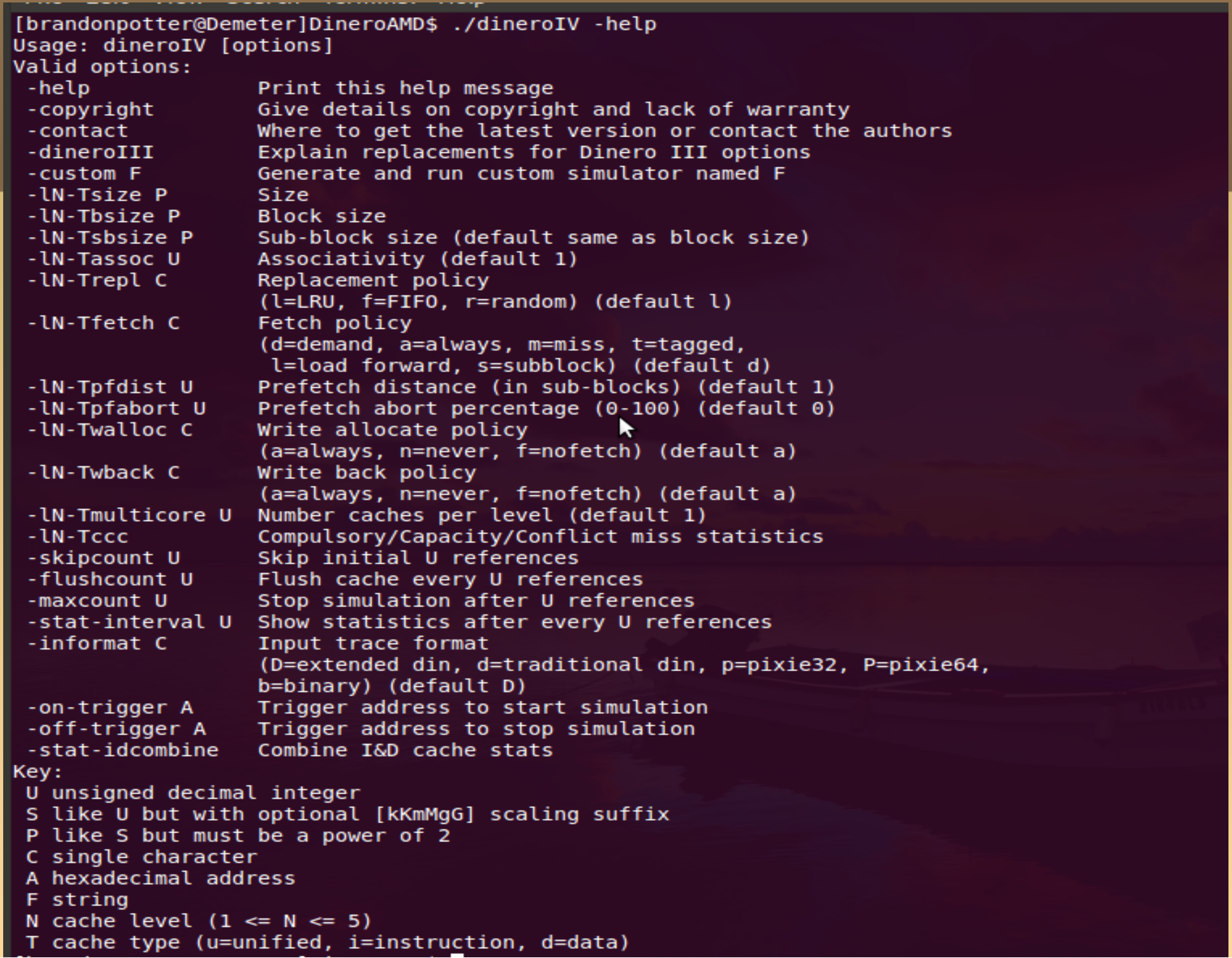
Semester Project

- Your semester project requires work on something architecture related.
- DineroIV would be an “easy” project to work on.
  - Majority of work is already done.
  - It has relatively small code base.
- DineroIV needs verification!
  - C programming
  - Understand cache policy (write-through/write-back)
  - Understand coherency (directory/MESI)
  - Write cases to try to break DineroIV!

DineroIV Options

- Command Line Options (CLO)
- Many parameters can be set:
  - Cache size
  - Block/subblock size
  - Associativity
  - Replacement policy
  - Prefetching
  - Write allocation policy
  - Multicore
- Try “./dineroIV -help”

DineroIV Options

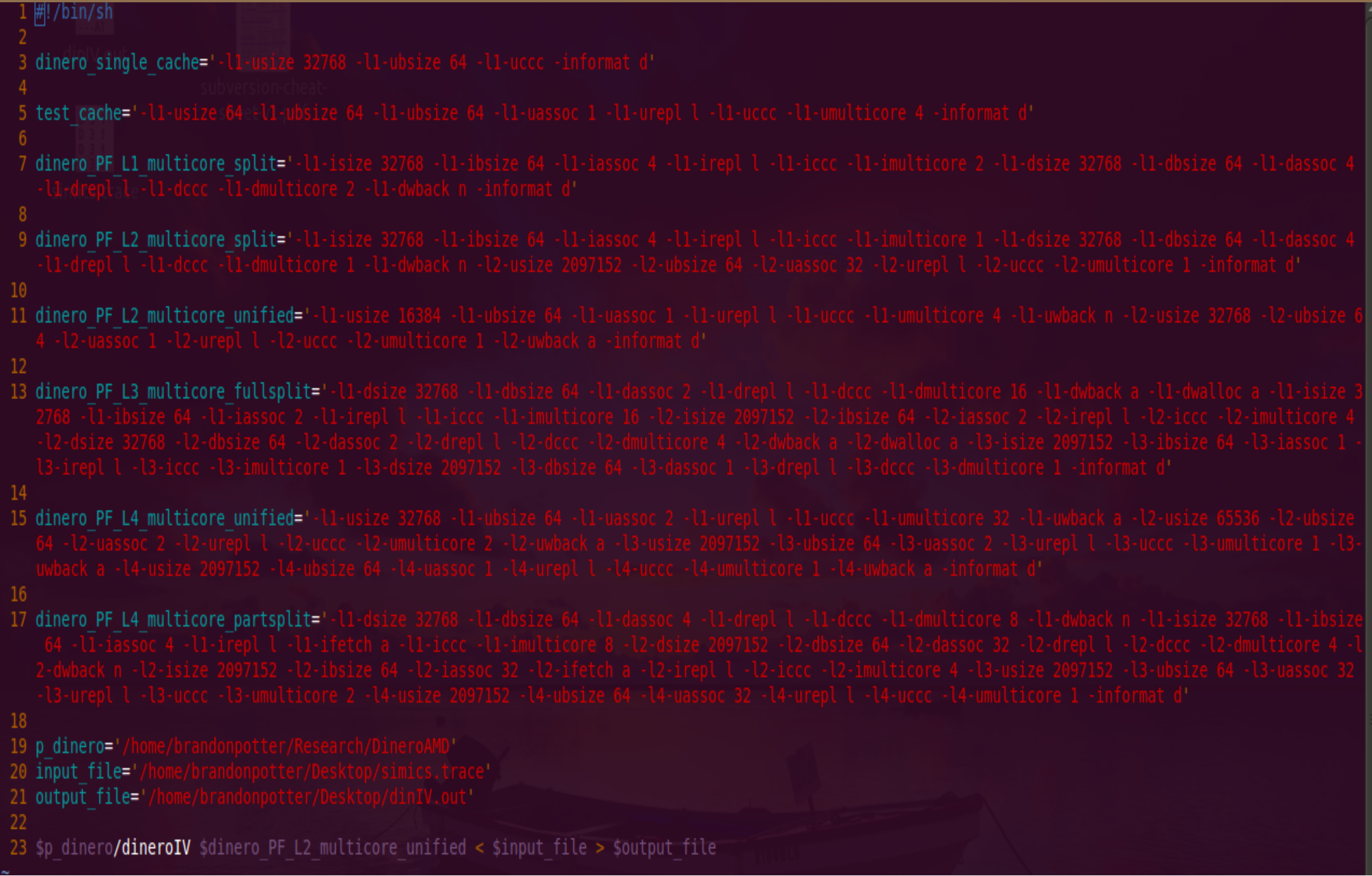


```
[brandonpotter@Demeter]DineroAMD$ ./dineroIV -help
Usage: dineroIV [options]
Valid options:
 -help                Print this help message
 -copyright           Give details on copyright and lack of warranty
 -contact             Where to get the latest version or contact the authors
 -dineroIII           Explain replacements for Dinero III options
 -custom F            Generate and run custom simulator named F
 -lN-Tsize P          Size
 -lN-Tbsize P         Block size
 -lN-Tsbsize P        Sub-block size (default same as block size)
 -lN-Tassoc U         Associativity (default 1)
 -lN-Trepl C          Replacement policy
                      (l=LRU, f=FIFO, r=random) (default l)
 -lN-Tfetch C         Fetch policy
                      (d=demand, a=always, m=miss, t=tagged,
                       l=load forward, s=subblock) (default d)
 -lN-Tpfdist U        Prefetch distance (in sub-blocks) (default 1)
 -lN-Tpfabort U       Prefetch abort percentage (0-100) (default 0)
 -lN-Twalloc C        Write allocate policy
                      (a=always, n=never, f=nofetch) (default a)
 -lN-Twback C         Write back policy
                      (a=always, n=never, f=nofetch) (default a)
 -lN-Tmulticore U     Number caches per level (default 1)
 -lN-Tccc             Compulsory/Capacity/Conflict miss statistics
 -skipcount U         Skip initial U references
 -flushcount U        Flush cache every U references
 -maxcount U          Stop simulation after U references
 -stat-interval U     Show statistics after every U references
 -informat C          Input trace format
                      (D=extended din, d=traditional din, p=pixie32, P=pixie64,
                      b=binary) (default D)
 -on-trigger A        Trigger address to start simulation
 -off-trigger A       Trigger address to stop simulation
 -stat-idcombine      Combine I&D cache stats
Key:
 U unsigned decimal integer
 S like U but with optional [kKmMgG] scaling suffix
 P like S but must be a power of 2
 C single character
 A hexadecimal address
 F string
 N cache level (1 <= N <= 5)
 T cache type (u=unified, i=instruction, d=data)
```

DineroIV Invocation

- Two methods:
  - ./dineroIV -l1-usize 32768 -l1-ubsize -l1-uccc -informat d < trace.trace > dinIV.out
  - cd ~/dineroIV/script; ./runParams

DineroIV Invocation

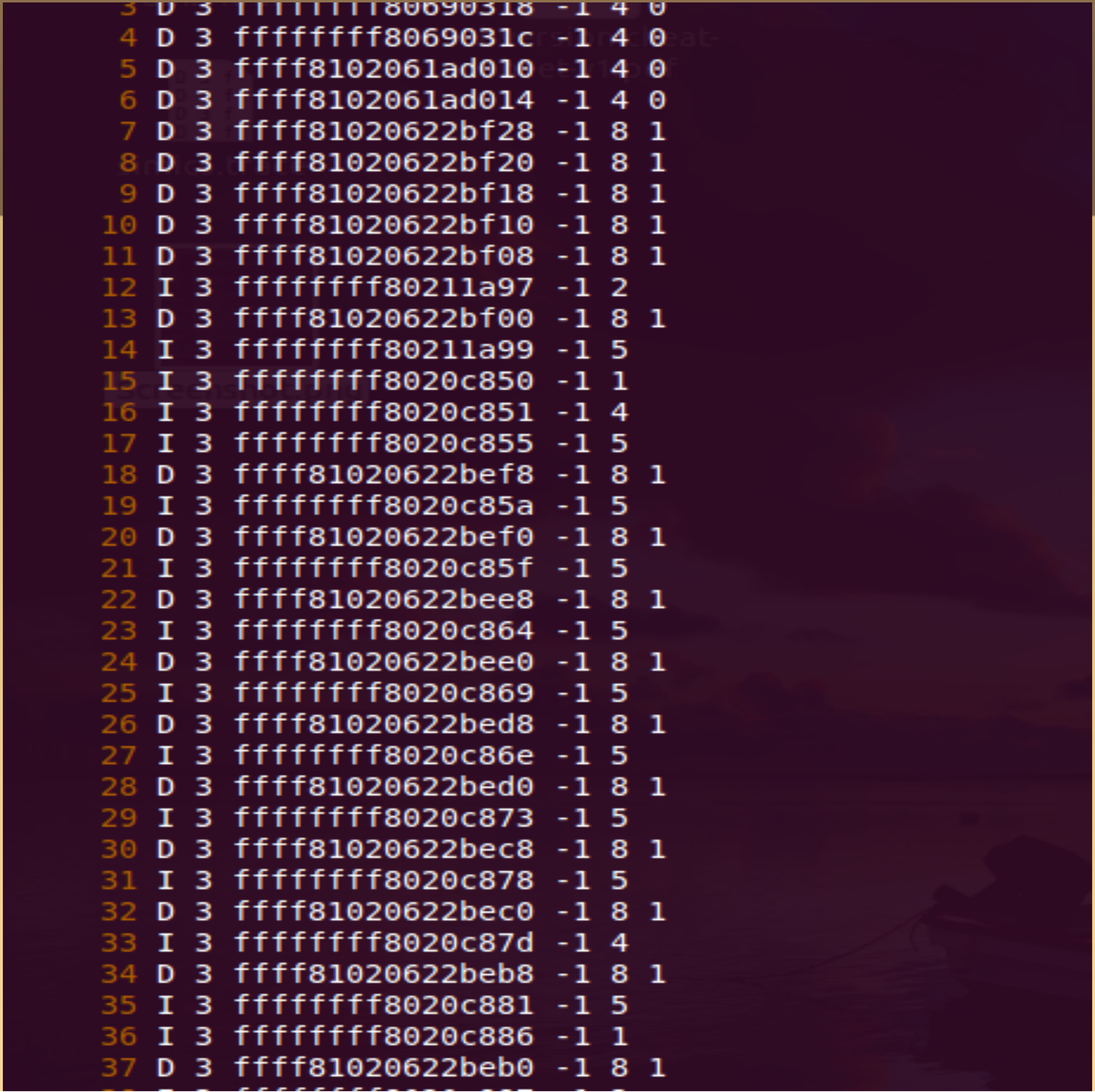


```
#!/bin/sh

dinero_single_cache='-l1-usize 32768 -l1-ubsize 64 -l1-uccc -informat d'

test_cache='-l1-usize 64 -l1-ubsize 64 -l1-ubsize 64 -l1-uassoc 1 -l1-urepl l -l1-uccc -l1-umulticore 4 -informat d'

dinero_PF_L1_multicore_split='-l1-isize 32768 -l1-ibsize 64 -l1-iassoc 4 -l1-irepl l -l1-iccc -l1-imulticore 2 -l1-dsize 32768 -l1-dbsize 64 -l1-dassoc 4 -l1-drepl l -l1-dccc -l1-dmulticore 2 -l1-dwback n -informat d'

dinero_PF_L2_multicore_split='-l1-isize 32768 -l1-ibsize 64 -l1-iassoc 4 -l1-irepl l -l1-iccc -l1-imulticore 1 -l1-dsize 32768 -l1-dbsize 64 -l1-dassoc 4 -l1-drepl l -l1-dccc -l1-dmulticore 1 -l1-dwback n -l2-usize 2097152 -l2-ubsize 64 -l2-uassoc 32 -l2-urepl l -l2-uccc -l2-umulticore 1 -informat d'

dinero_PF_L2_multicore_unified='-l1-usize 16384 -l1-ubsize 64 -l1-uassoc 1 -l1-urepl l -l1-uccc -l1-umulticore 4 -l1-uwback n -l2-usize 32768 -l2-ubsize 64 -l2-uassoc 1 -l2-urepl l -l2-uccc -l2-umulticore 1 -l2-uwback a -informat d'

dinero_PF_L3_multicore_fullsplit='-l1-dsize 32768 -l1-dbsize 64 -l1-dassoc 2 -l1-drepl l -l1-dccc -l1-dmulticore 16 -l1-dwback a -l1-dwalloc a -l1-isize 32768 -l1-ibsize 64 -l1-iassoc 2 -l1-irepl l -l1-iccc -l1-imulticore 16 -l2-isize 2097152 -l2-ibsize 64 -l2-iassoc 2 -l2-irepl l -l2-iccc -l2-imulticore 4 -l2-dsize 32768 -l2-dbsize 64 -l2-dassoc 2 -l2-drepl l -l2-dccc -l2-dmulticore 4 -l2-dwback a -l2-dwalloc a -l3-isize 2097152 -l3-ibsize 64 -l3-iassoc 1 -l3-irepl l -l3-iccc -l3-imulticore 1 -l3-dsize 2097152 -l3-dbsize 64 -l3-dassoc 1 -l3-drepl l -l3-dccc -l3-dmulticore 1 -informat d'

dinero_PF_L4_multicore_unified='-l1-usize 32768 -l1-ubsize 64 -l1-uassoc 2 -l1-urepl l -l1-uccc -l1-umulticore 32 -l1-uwback a -l2-usize 65536 -l2-ubsize 64 -l2-uassoc 2 -l2-urepl l -l2-uccc -l2-umulticore 2 -l2-uwback a -l3-usize 2097152 -l3-ubsize 64 -l3-uassoc 2 -l3-urepl l -l3-uccc -l3-umulticore 1 -l3-uwback a -l4-usize 2097152 -l4-ubsize 64 -l4-uassoc 1 -l4-urepl l -l4-uccc -l4-umulticore 1 -l4-uwback a -informat d'

dinero_PF_L4_multicore_partsplit='-l1-dsize 32768 -l1-dbsize 64 -l1-dassoc 4 -l1-drepl l -l1-dccc -l1-dmulticore 8 -l1-dwback n -l1-isize 32768 -l1-ibsize 64 -l1-iassoc 4 -l1-irepl l -l1-ifetch a -l1-iccc -l1-imulticore 8 -l2-dsize 2097152 -l2-dbsize 64 -l2-dassoc 32 -l2-drepl l -l2-dccc -l2-dmulticore 4 -l2-dwback n -l2-isize 2097152 -l2-ibsize 64 -l2-iassoc 32 -l2-ifetch a -l2-irepl l -l2-iccc -l2-imulticore 4 -l3-usize 2097152 -l3-ubsize 64 -l3-uassoc 32 -l3-urepl l -l3-uccc -l3-umulticore 2 -l4-usize 2097152 -l4-ubsize 64 -l4-uassoc 32 -l4-urepl l -l4-uccc -l4-umulticore 1 -informat d'

p_dinero='/home/brandonpotter/Research/DineroAMD'
input_file='/home/brandonpotter/Desktop/simics.trace'
output_file='/home/brandonpotter/Desktop/dinIV.out'

$p_dinero/dineroIV $dinero_PF_L2_multicore_unified < $input_file > $output_file
```

Input Trace File

- As mentioned previously, DineroIV is trace driven!
- From left to right:
  - Instruction/data cache
  - Core number (zero based)
  - Virtual address
  - Process id
  - Size
  - Read/write

Input Trace File

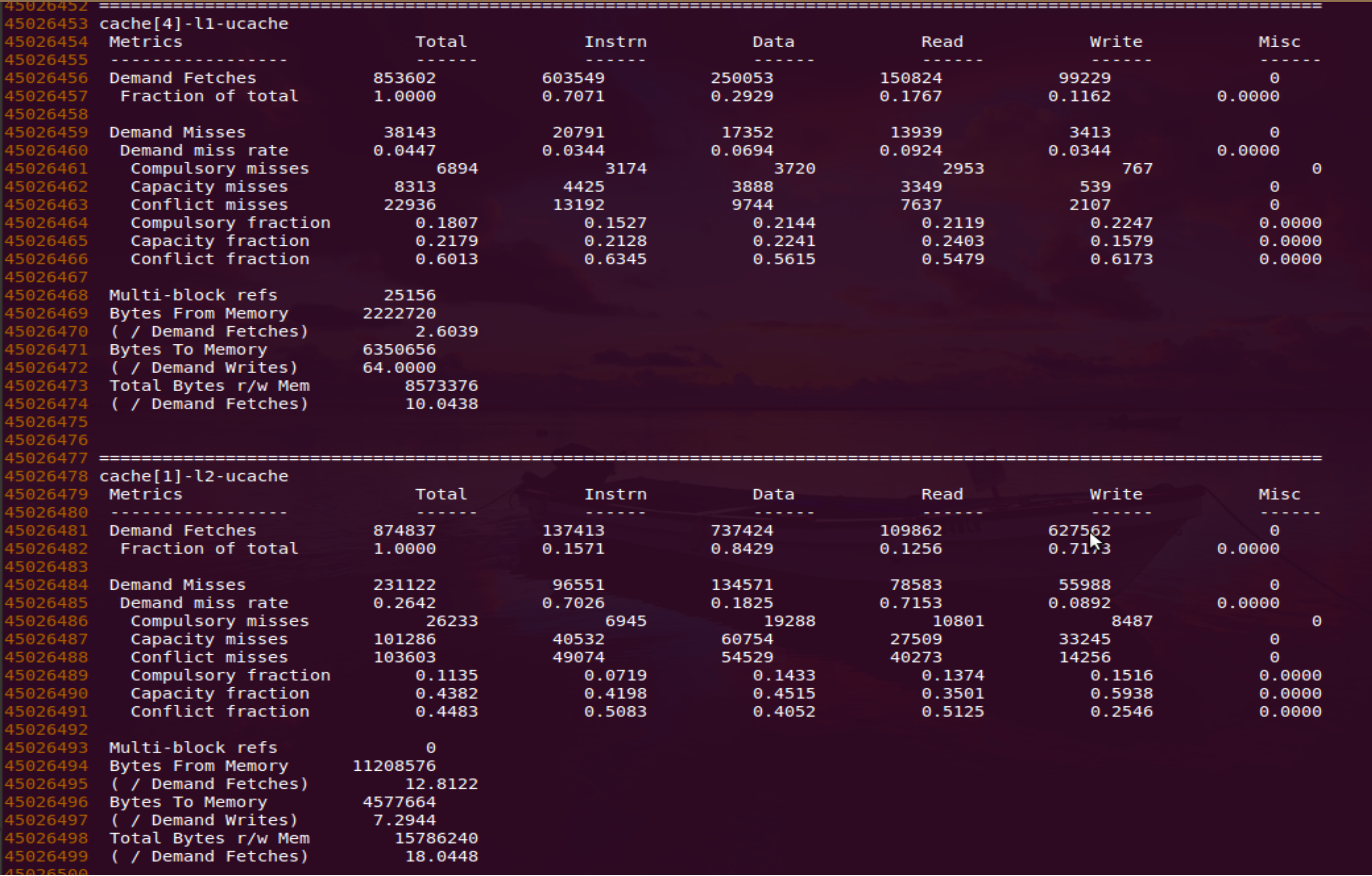


```
D 3 ffffffff80690318 -1 4 0
D 3 ffffffff8069031c -1 4 0
D 3 ffff8102061ad010 -1 4 0
D 3 ffff8102061ad014 -1 4 0
D 3 ffff81020622bf28 -1 8 1
D 3 ffff81020622bf20 -1 8 1
D 3 ffff81020622bf18 -1 8 1
D 3 ffff81020622bf10 -1 8 1
D 3 ffff81020622bf08 -1 8 1
I 3 ffffffff80211a97 -1 2
D 3 ffff81020622bf00 -1 8 1
I 3 ffffffff80211a99 -1 5
I 3 ffffffff8020c850 -1 1
I 3 ffffffff8020c851 -1 4
I 3 ffffffff8020c855 -1 5
D 3 ffff81020622bef8 -1 8 1
I 3 ffffffff8020c85a -1 5
D 3 ffff81020622bef0 -1 8 1
I 3 ffffffff8020c85f -1 5
D 3 ffff81020622bee8 -1 8 1
I 3 ffffffff8020c864 -1 5
D 3 ffff81020622bee0 -1 8 1
I 3 ffffffff8020c869 -1 5
D 3 ffff81020622bed8 -1 8 1
I 3 ffffffff8020c86e -1 5
D 3 ffff81020622bed0 -1 8 1
I 3 ffffffff8020c873 -1 5
D 3 ffff81020622bec8 -1 8 1
I 3 ffffffff8020c878 -1 5
D 3 ffff81020622bec0 -1 8 1
I 3 ffffffff8020c87d -1 4
D 3 ffff81020622beb8 -1 8 1
I 3 ffffffff8020c881 -1 5
I 3 ffffffff8020c886 -1 1
D 3 ffff81020622beb0 -1 8 1
```

DineroIV Output

- Each cache will spit out information regarding miss statistics
- Cache ids are a little confusing
  - Ids do not correspond to core numbers – BEWARE!
  - Ids must be unique within DineroIV
- Core interconnect connects all the caches in logically symmetric fashion from left to right
  - Core 0 L1$ will print first followed by Core 1, etc...

DineroIV Output

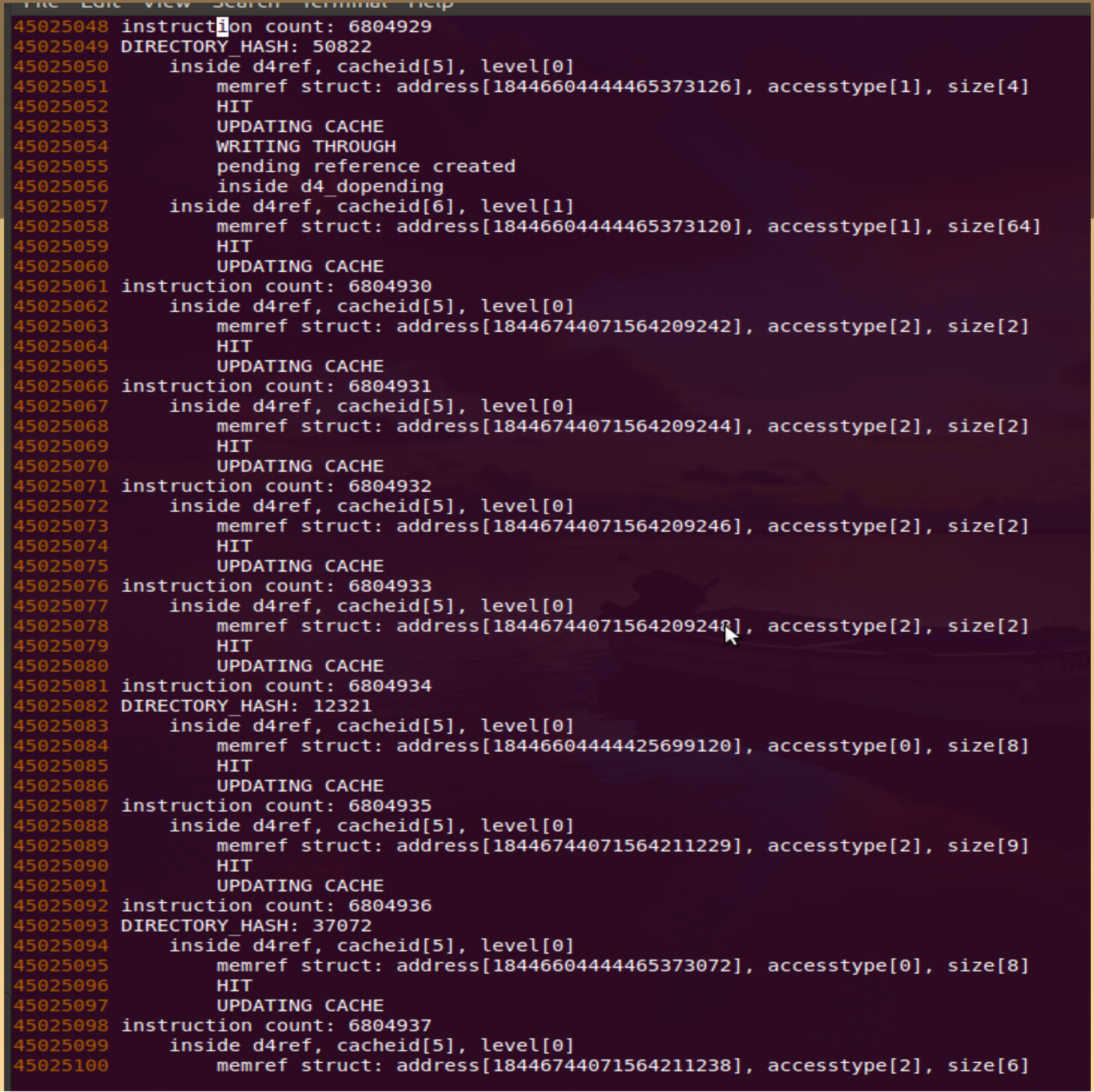


```
==================================================================================================================
cache[4]-l1-ucache
 Metrics                      Total          Instrn          Data           Read          Write           Misc
 -----------------           ------          ------         ------         ------         ------         ------
 Demand Fetches              853602         603549         250053         150824          99229              0
  Fraction of total          1.0000         0.7071         0.2929         0.1767         0.1162         0.0000

 Demand Misses                38143          20791          17352          13939           3413              0
  Demand miss rate           0.0447         0.0344         0.0694         0.0924         0.0344         0.0000
   Compulsory misses            6894           3174           3720           2953            767              0
   Capacity misses             8313           4425           3888           3349            539              0
   Conflict misses            22936          13192           9744           7637           2107              0
   Compulsory fraction          0.1807         0.1527         0.2144         0.2119         0.2247         0.0000
   Capacity fraction            0.2179         0.2128         0.2241         0.2403         0.1579         0.0000
   Conflict fraction            0.6013         0.6345         0.5615         0.5479         0.6173         0.0000

 Multi-block refs             25156
 Bytes From Memory          2222720
 ( / Demand Fetches)            2.6039
 Bytes To Memory            6350656
 ( / Demand Writes)         64.0000
 Total Bytes r/w Mem           8573376
 ( / Demand Fetches)           10.0438


==================================================================================================================
cache[1]-l2-ucache
 Metrics                      Total          Instrn          Data           Read          Write           Misc
 -----------------           ------          ------         ------         ------         ------         ------
 Demand Fetches              874837         137413         737424         109862         627562              0
  Fraction of total          1.0000         0.1571         0.8429         0.1256         0.7173         0.0000

 Demand Misses               231122          96551         134571          78583          55988              0
  Demand miss rate           0.2642         0.7026         0.1825         0.7153         0.0892         0.0000
   Compulsory misses           26233           6945          19288          10801           8487              0
   Capacity misses           101286          40532          60754          27509          33245              0
   Conflict misses           103603          49074          54529          40273          14256              0
   Compulsory fraction          0.1135         0.0719         0.1433         0.1374         0.1516         0.0000
   Capacity fraction            0.4382         0.4198         0.4515         0.3501         0.5938         0.0000
   Conflict fraction            0.4483         0.5083         0.4052         0.5125         0.2546         0.0000

 Multi-block refs                 0
 Bytes From Memory         11208576
 ( / Demand Fetches)           12.8122
 Bytes To Memory            4577664
 ( / Demand Writes)          7.2944
 Total Bytes r/w Mem          15786240
 ( / Demand Fetches)           18.0448
```

DineroIV Debug

- I created a few printf() debug statements to help me
- Enabled/disabled inside d4.h
  - Search for D4DEBUG
  - 1 – on
  - 0 – off
- Finding the printf() calls inside the code is a good idea to familiarize yourself with the inner workings; they highlight important areas
- Warning: creates EXTREMELY long output

DineroIV Debug



```
instruction count: 6804929
DIRECTORY_HASH: 50822
    inside d4ref, cacheid[5], level[0]
        memref struct: address[18446604444465373126], accesstype[1], size[4]
        HIT
        UPDATING CACHE
        WRITING THROUGH
        pending reference created
        inside d4_dopending
    inside d4ref, cacheid[6], level[1]
        memref struct: address[18446604444465373120], accesstype[1], size[64]
        HIT
        UPDATING CACHE
instruction count: 6804930
    inside d4ref, cacheid[5], level[0]
        memref struct: address[18446744071564209242], accesstype[2], size[2]
        HIT
        UPDATING CACHE
instruction count: 6804931
    inside d4ref, cacheid[5], level[0]
        memref struct: address[18446744071564209244], accesstype[2], size[2]
        HIT
        UPDATING CACHE
instruction count: 6804932
    inside d4ref, cacheid[5], level[0]
        memref struct: address[18446744071564209246], accesstype[2], size[2]
        HIT
        UPDATING CACHE
instruction count: 6804933
    inside d4ref, cacheid[5], level[0]
        memref struct: address[18446744071564209248], accesstype[2], size[2]
        HIT
        UPDATING CACHE
instruction count: 6804934
DIRECTORY_HASH: 12321
    inside d4ref, cacheid[5], level[0]
        memref struct: address[18446604444425699120], accesstype[0], size[8]
        HIT
        UPDATING CACHE
instruction count: 6804935
    inside d4ref, cacheid[5], level[0]
        memref struct: address[18446744071564211229], accesstype[2], size[9]
        HIT
        UPDATING CACHE
instruction count: 6804936
DIRECTORY_HASH: 37072
    inside d4ref, cacheid[5], level[0]
        memref struct: address[18446604444465373072], accesstype[0], size[8]
        HIT
        UPDATING CACHE
instruction count: 6804937
    inside d4ref, cacheid[5], level[0]
        memref struct: address[18446744071564211238], accesstype[2], size[6]
```

Further Information

- Source code:
  - svn co http://csrl.unt.edu/svn/tools/DineroIV_MC/dineroIV –username=guest
  - Password: guest2011
- CSRL DineroIV webpage:
  - http://csrl.unt.edu/dineroIVmc/
  - Installation instructions

Contact

- Email:
  - brandonpotter[at]my[dot]unt[dot]edu
- Office:
  - F232
  - Between 10:00 and 17:00
- Dr. Kavi:
  - He is a great resource as he is willing to talk to students almost always. If you have general architecture questions, go talk to him!

Questions