

# Computer Architecture - Homework 3 Report

Student ID: B11901110

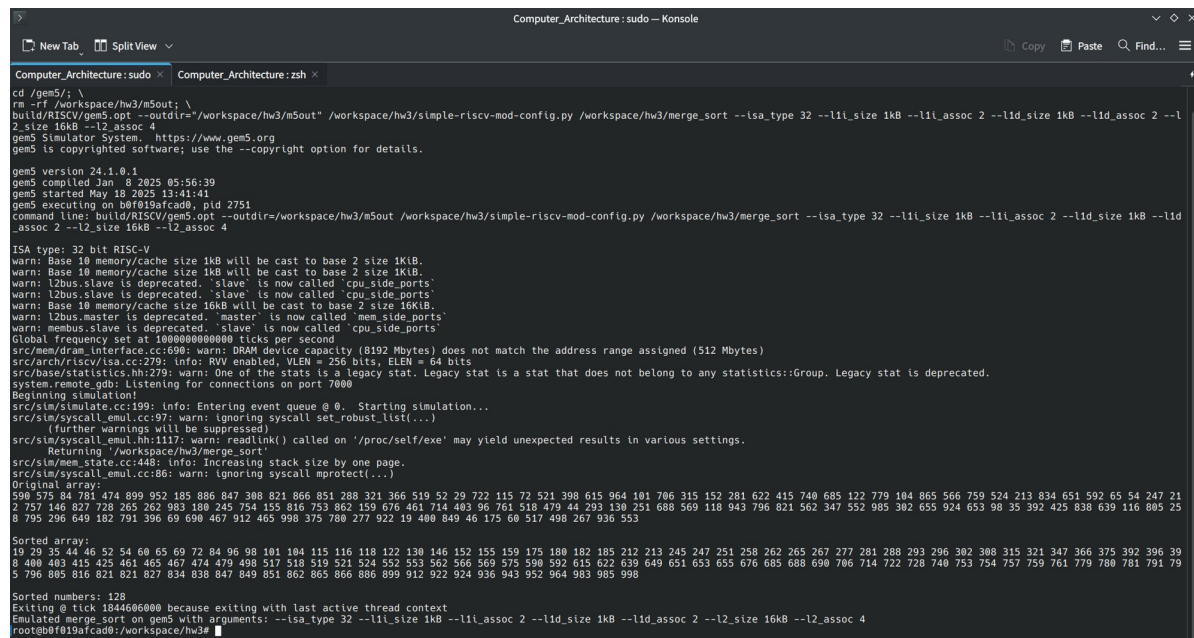
Name: 陳璿吉

## Part1. Add new components to gem5 config.

- 1-1: Implement L2 cache in gem5 config
- 1-2: Draw and analyze the program results based on the config

Part1-1. Show screenshots of your program running successfully (refer to P.11) (15%)

The result is shown in the following figure.



```
Computer_Architecture: sudo - Konsole
Computer_Architecture: sudo x Computer_Architecture: zsh x
cd /gem5/; \
rm -rf /workspace/hw3/m5out; \
build/RISCV/gem5.opt --outdir="/workspace/hw3/m5out" /workspace/hw3/simple-riscv-mod-config.py /workspace/hw3/merge_sort --isa_type 32 --l1l_size 1kB --l1l_assoc 2 --l1d_size 1kB --l1d_assoc 2 --l2_size 16kB --l2_assoc 4
gem5 Simulator System. https://www.gem5.org
gem5 is copyrighted software; use the --copyright option for details.

gem5 version 24.1.0.1
gem5 compiled Jan  8 2025 05:56:39
gem5 started May 18 2025 12:41:41
gem5 executing on b0f019afcad0, pid 2751
command line: build/RISCV/gem5.opt --outdir=/workspace/hw3/m5out /workspace/hw3/simple-riscv-mod-config.py /workspace/hw3/merge_sort --isa_type 32 --l1l_size 1kB --l1l_assoc 2 --l1d_size 1kB --l1d_assoc 2 --l2_size 16kB --l2_assoc 4

ISA type: 32 bit RISCV-V
warn: Base 10 memory/cache size 1kB will be cast to base 2 size 1KiB.
warn: Base 10 memory/cache size 1kB will be cast to base 2 size 1KiB.
warn: l2bus.slave is deprecated. 'slave' is now called 'cpu_side_ports'
warn: l2bus.slave is deprecated. 'slave' is now called 'cpu_side_ports'
warn: Base 10 memory/cache size 16kB will be cast to base 2 size 16KiB.
warn: l2bus.master is deprecated. 'master' is now called 'mem_side_ports'
warn: membus.slave is deprecated. 'slave' is now called 'cpu_side_ports'
Global frequency set at 100000000000 ticks per second
src/mm/dram_interface.cc:690: warn: DRAM device capacity (8192 Mbytes) does not match the address range assigned (512 Mbytes)
src/arch/riscv/isa.cc:270: info: RVV enabled, VLEN = 256 bits, ELEN = 64 bits
src/base/statistics.hh:279: warn: One of the stats is a legacy stat. Legacy stat is deprecated.
system.remote.gdb: Listening for connections on port 7000
Beginning simulation!
src/sim/simulate.cc:199: info: Entering event queue @ 0. Starting simulation...
src/sim/syscall_emul.cc:97: warn: Ignoring syscall set_robust_list(...)
(further warnings will be suppressed)
src/sim/syscall_emul.hh:1117: warn: readlink() called on '/proc/self/exe' may yield unexpected results in various settings.
Returning /workspace/hw3/merge_sort
src/sim/mem_state.cc:449: info: Increasing stack size by one page.
src/sim/syscall_emul.cc:86: warn: Ignoring syscall mprotect(...)
Original array:
590 575 84 701 474 899 952 185 886 847 308 821 866 851 288 321 366 519 52 29 722 115 72 521 398 615 964 101 786 315 152 281 622 415 740 685 122 779 184 865 566 759 524 213 834 651 592 65 54 247 21
2 757 146 827 728 265 262 983 180 245 754 155 816 753 862 159 676 461 714 403 96 761 518 479 44 293 130 251 688 569 118 943 796 821 562 347 552 985 302 655 924 653 98 35 392 425 838 639 116 805 25
8 795 296 649 182 791 396 69 698 467 912 465 998 375 780 277 922 19 480 849 46 175 60 517 498 267 936 553

Sorted array:
19 29 35 44 46 52 54 60 65 69 72 84 96 98 101 104 115 116 118 122 130 146 152 155 159 175 180 182 185 212 213 245 247 251 258 262 265 267 277 281 288 293 296 302 308 315 321 347 366 375 392 396 39
0 400 403 415 425 461 465 467 474 479 490 517 518 519 521 524 552 553 562 566 569 575 590 592 615 622 639 649 651 653 655 676 685 688 690 706 714 722 728 740 753 754 757 759 761 779 780 781 791 79
5 796 805 816 821 821 827 834 838 847 849 851 862 865 866 886 899 912 922 924 936 943 952 964 983 985 998

Sorted numbers: 128
Exiting @ tick 1844686000 because exiting with last active thread context
Emulated merge_sort on gem5 with arguments: --isa_type 32 --l1l_size 1kB --l1l_assoc 2 --l1d_size 1kB --l1d_assoc 2 --l2_size 16kB --l2_assoc 4
root@b0f019afcad0:/workspace/hw3#
```

Part1-1. Show screenshots of your program and cache summary (refer to P.12) (15%)

<program summary>

```
Program summary
-----
simulated time      | 0.001845 s
simulated tick      | 1,844,606,000 ticks
total Inst.         | 378,971 instructions
total cycle         | 1,844,606 cycles
CPI                 | 4.863852
IPC                 | 0.205598
Int-Inst. count     | 376,040 instructions
Load-Inst. count    | 80,758 instructions
Store-Inst. count    | 40,786 instructions
Vector-Inst. count  | 0 instructions
```

<L1 cache summary>

```
L1-Instruction-Cache summary
-----
$L1-I hit count     | 442,397 counts
$L1-I miss count    | 17,213 counts
$L1-I access count  | 459,610 counts
$L1-I miss rate     | 3.75% miss rate
L1-I assoc          | 2
L1-I size           | 1024

L1-Data-Cache summary
-----
$L1-D hit count     | 114,824 counts
$L1-D miss count    | 6,682 counts
$L1-D access count  | 121,506 counts
$L1-D miss rate     | 5.50% miss rate
L1-D assoc          | 2
L1-D size           | 1024
```

<L2 cache summary>

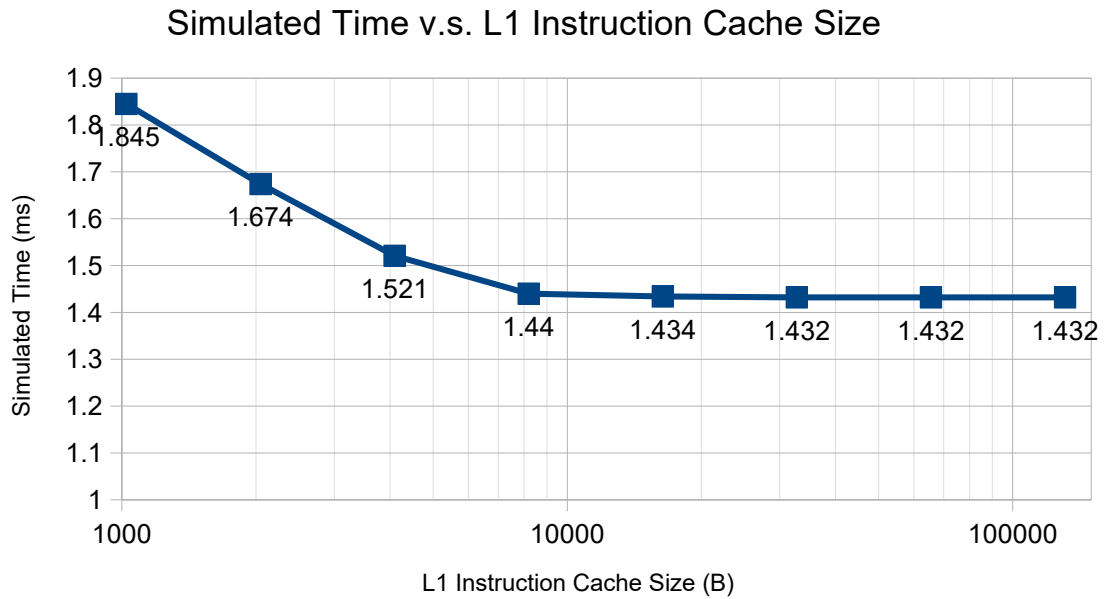
L2-Cache summary	
-----	
\$L2 hit count	22,493 counts
\$L2 miss count	1,408 counts
\$L2 access count	23,901 counts
\$L2 miss rate	5.89% miss rate
L2 assoc	4
L2 size	16384

Part1-2. Draw a graph based on the different hardware config (refer to P.13-14). (40%)

Here is the default setting:

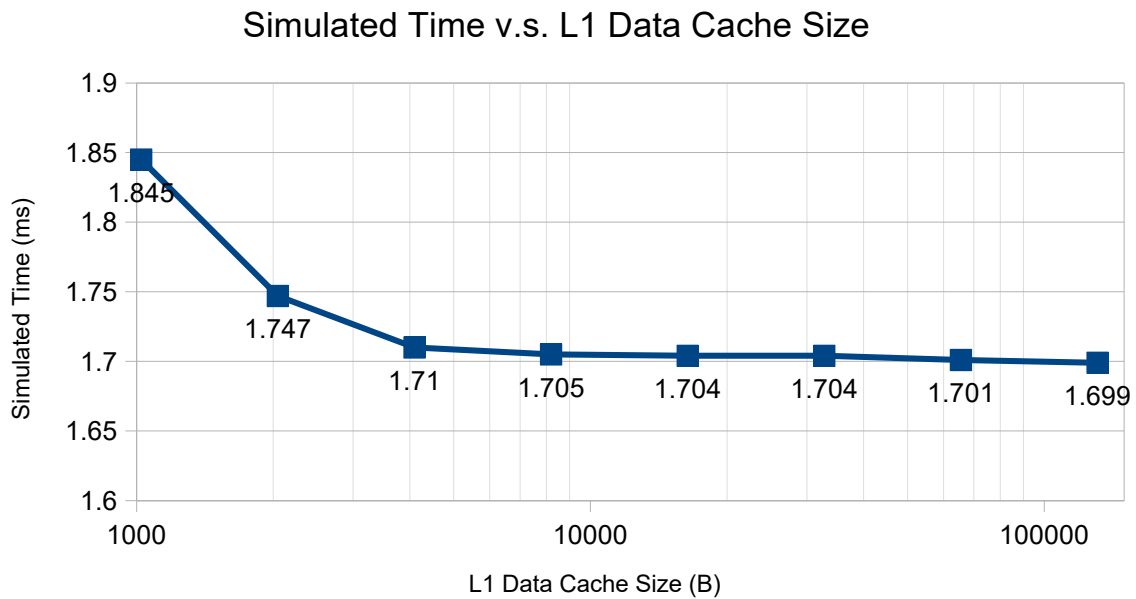
- L1 Instruction Cache Size: 1 kB
- L1 Instruction Cache Associativity: 2
- L1 Data Cache Size: 1 kB
- L1 Data Cache Associativity: 2
- L2 Cache Size: 16 kB
- L2 Cache Associativity: 4

<L1-I cache size>



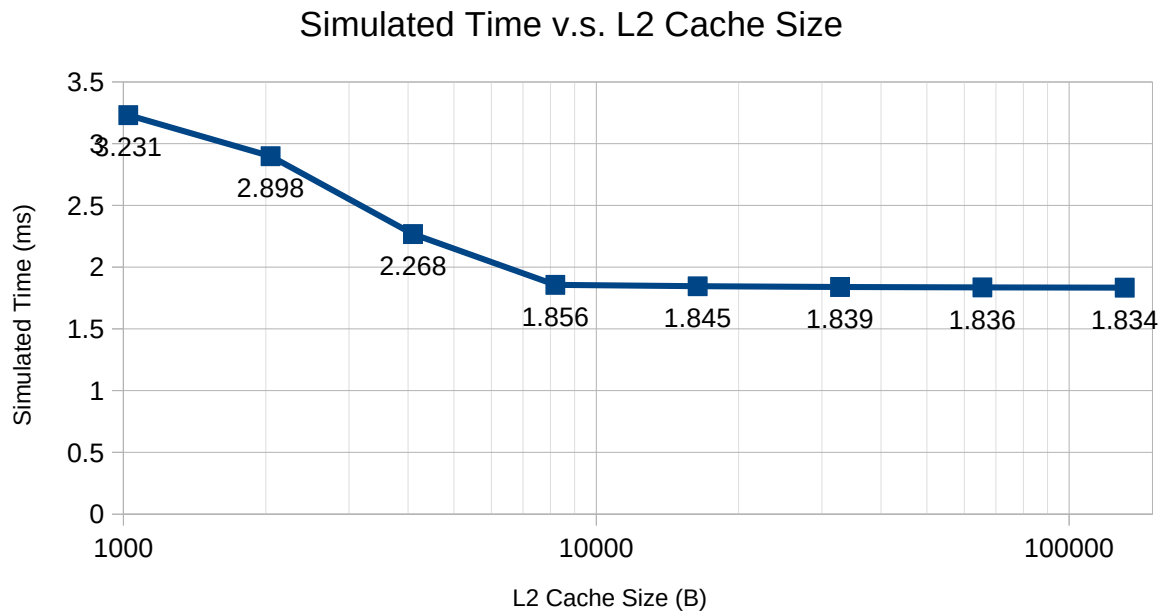
As we increase L1 instruction cache size, we can store more instructions in the cache. The miss rate decreases, but the effect diminished as L1 instruction cache size became larger than 8 kB.

<L1-D cache size>



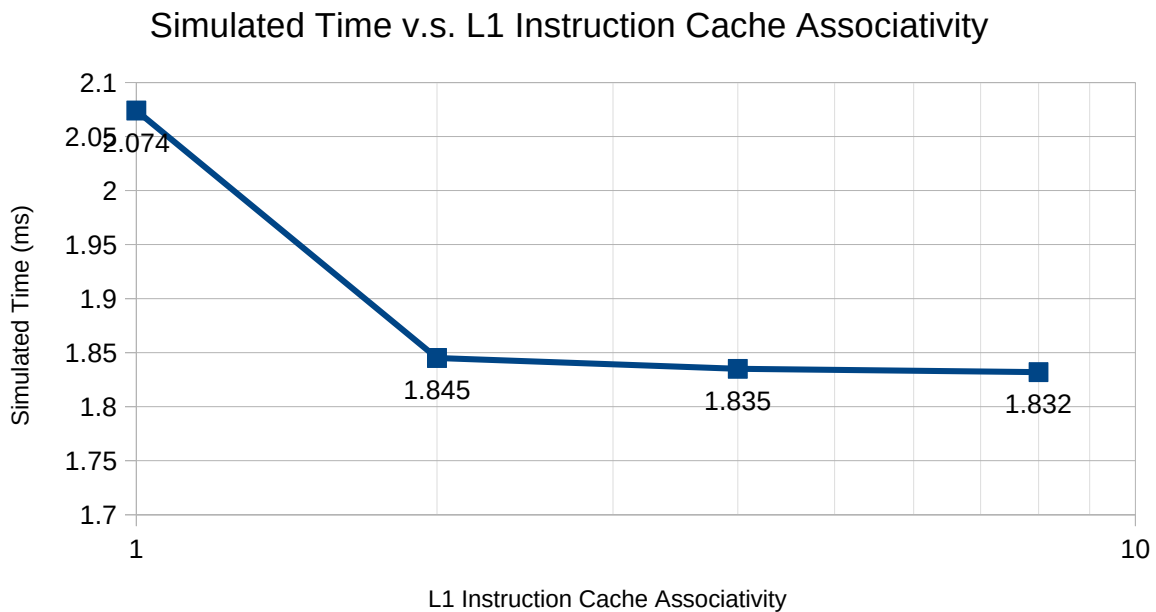
Similar to the case in L1 instruction cache, increasing the size of L1 data cache also results in the decrease of running time. Also, the effect diminished after the size was over 8 kB. However, compared to increasing the size of L1 instruction cache, increasing L1 data cache size has less significant effect.

<L2 cache size>



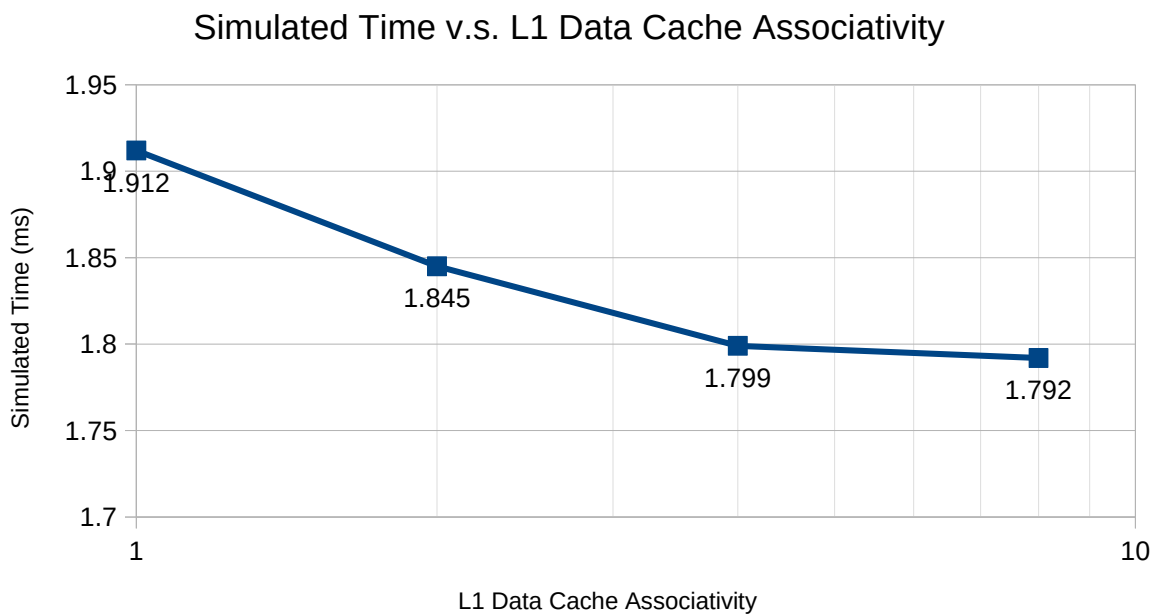
In this program, the default L2 cache size is 16 kB. Once again, the execution time decreases as L2 cache size increases. However, the effect diminished after the cache size exceeded 16 kB.

<L1-I associativity>



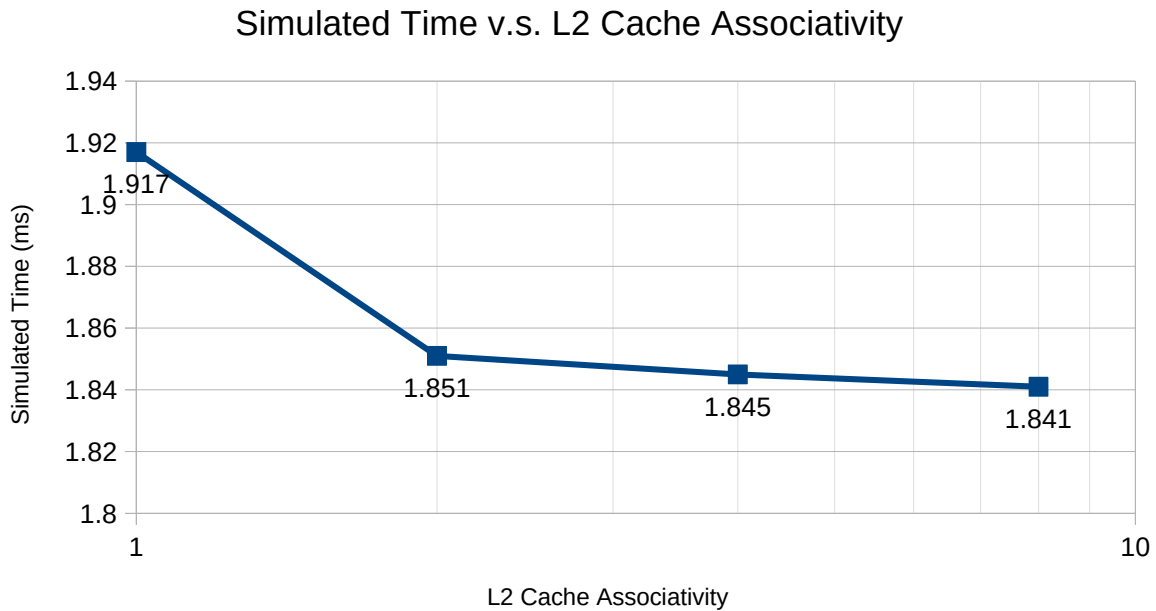
As the graph illustrated, increasing associativity of L1 instruction cache can reduce the execution time of the program. However, the effect diminished as the associativity exceeded 2.

<L1-D associativity>



Similar to the trend observed in L1 instruction cache, the execution time decreases as we increase the associativity of L1 data cache. As opposite to instruction cache, increasing L1 data cache associativity has more effect compared to increasing L1 instruction cache associativity.

<L2 associativity>



Once again, we observed similar trend as in the previous cases. In particular, the effect is similar to increasing L1 instruction cache associativity.

**Part2. Analyze and find the optimal config based on the application**

- **Cache Size: 1KB ~ 128KB**
- **Associativity: 1 ~ 8**

Part2-1. Find the optimal config settings for this program. (10%)

Config	Cache size (KB)	Associativity
L1-I cache	32	8
L1-D cache	64	8
L2 cache	1	1

Performance	Score
23	12

※ Save your optimal config in gem5\_args.conf



Part2-2. Explain why this config achieves optimal performance ? (20%)

By the nature of the L1 cache, it is the most frequently accessed component of a CPU. Therefore, increasing the size of L1 cache can improve the performance dramatically.

However, which one accounts for more improvement, L1 instruction cache or L1 data cache? This is related to the algorithm --- merge sort. Merge sort is a divide-and-conquer algorithm, which means that there are a lot of recursive calls. A recursive call is realized by unconditional jump in assembly language. Intuitively speaking, the program needs to fetch more instructions to perform recursive call. Therefore, we need to increase the size of the L1 instruction cache. In this case, the number of ticks stabilized after the L1 cache size exceeded 32 kB.

When the recursion of merge sort reaches the bottom (leaves), we need to perform comparison of two nodes, and put them in appropriate addresses. In assembly language, accessing data and making comparison means we need more space for data to reduce miss rate. Once we have enough space for instructions, increasing L1 data cache size can further improve the performance. To make sure the performance is optimized, we set the associativity to 8, the maximum allowed in this assignment.

What about the effect of L2 cache? By the nature of L2 cache, and the data of program execution, L2 cache access (23901 times) is significantly less than L1 cache (459610 and 121506 times). Once we have sufficient space for L1 caches, then L2 cache becomes negligible in this case. Therefore, we set the size of L2 cache to 1 kB, and the associativity to 1.