Computer Architecture Lab

12. cache 2

MPLAB Media Processing Laboratory

컴퓨터 공학과 이성원 교수님 연구실

SPEC95 Benchmarks

- SPEC 95 (Standard Performance Evaluation Corporation)

CINT95 benchmarks:

go	- Artificial intelligence; plays the game of "Go"
m88ksim	- Moto 88K Chip simulator; runs test program
gcc	- New version of GCC; builds SPARC code
compress	- Compresses and decompresses file in memory
li	- LISP interpreter
ijpeg	- Graphic compression and decompression
perl	- Manipulates strings (anagrams) and prime numbers in Perl
vortex	- A database program

CFP95 benchmarks:

tomcatv	-A mesh-generation program
swim	-Shallow water model with 513 x 513 grid
su2cor	-Quantum physics; Monte Carlo simulation
hydro2d	-Astrophysics; Hydrodynamical Navier Stokes equations
mgrid	-Multi-grid solver in 3D potential field
applu	-Parabolic/elliptic partial differential equations
turb3d	-Simulates isotropic, homogeneous turbulence in a cube
apsi	-Solves problems regarding temperature, wind, velocity and distribution of pollutants
fpppp	-Quantum chemistry
wave5	-Plasma physics; Electromagnetic particle simulation

SPEC95 Benchmarks

Table 1 – CINT95 Benchmarks

Benchmark	Application Area	Specific Task
li	Language interpreter	Lisp interpreter
vortex	Database	Builds and manipulates three interrelated databases
tomcatv	Geometric Translation	Generation of a two-dimensional vectorized mesh

- Change cache configuration
  - ✓ Unified cache / Separate cache
  - ✓ L1 cache size / L2 cache size
  - ✓ Large block size / Small block size
  - ✓ Direct-mapped / Set-associative

Write configuration file

- To seek help, type the following command
  - ✓ ./sim-cache –h
  - ✓ Cache setting : gedit ./config/mycache.cfg

The cache config parameter <config> has the following format:

<name>:<nsets>:<bsize>:<assoc>:<repl>

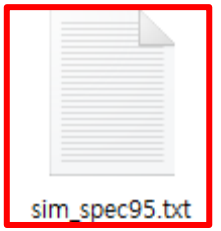
<name> - name of the cache being defined
<nsets> - number of sets in the cache
<bsize> - block size of the cache
<assoc> - associativity of the cache
<repl> - block replacement strategy, 'l'-LRU, 'f'-FIFO, 'r'-random

```
1 -cache:il1        il1:32:16:1:l
2
3 -cache:dl1        none                    #Data L1 cache
4 -cache:il2        il2:256:64:1:l          #Instruction L2 cache
5 -cachedl2         none                    #Data L2 cache
6 -tlb:itlb         none                    #Instruction TLB
7 -tlb:dtlb         none                    #Data TLB
```

[bytes]	[Cache Level 1]	[Cache Level2]
# of sets	32	256
Block size	16	64
Associativity	1	1
Total Cache size	512	16,384

How to simulate?

```
mpl@mpl-virtual-machine:~/simplesim-3.0$ ~/simplesim-3.0/sim-cache -config ~/sim
plesim-3.0/config/mycache.cfg bin/cc1.little.ss -O inputs/1stmt.i -o results/cc1
.out 2> traces/cc1.trace
```



- ./sim-cache –config <configfile path> <application path>

- li
  - ✓ ./sim-cache bin/li.little.ss inputs/train.lsp 2> traces/li.trace > results/li.out

- vortex
  - ✓ ./sim-cache bin/vortex.little.ss inputs/vortex.raw 2> traces/vortex.trace

- tomcatv
  - ✓ ./sim-cache bin/tomcatv.little.ss < inputs/tomcatv.in 2> traces/tomcatv.trace > results/tomcatv.out

Benchmark	Application Area	Specific Task
li	Language interpreter	Lisp interpreter
vortex	Database	Builds and manipulates three interrelated databases
tomcatv	Geometric Translation	Generation of a two-dimensional vectorized mesh

Example of result analysis

- Miss rate
  - ✓ check at .trace file
    (cd simplesim-3.0/traces -> vi or cat <name>.trace)


```
sim: ** simulation statistics **
sim_num_insn                362765 # total number of instructions executed
sim_num_refs                104761 # total number of loads and stores executed
sim_elapsed_time                 1 # total simulation time in seconds
sim_inst_rate           362765.0000 # simulation speed (in insts/sec)
il1.accesses                362765 # total number of accesses
il1.hits                    247961 # total number of hits
il1.misses                  114804 # total number of misses
il1.replacements            114772 # total number of replacements
il1.writebacks                   0 # total number of writebacks
il1.invalidations                0 # total number of invalidations
il1.miss_rate               0.3165 # miss rate (i.e., misses/ref)
il1.repl_rate               0.3164 # replacement rate (i.e., repls/ref)
il1.wb_rate                 0.0000 # writeback rate (i.e., wrbks/ref)
il1.inv_rate                0.0000 # invalidation rate (i.e., invs/ref)
il2.accesses                114804 # total number of accesses
il2.hits                    105675 # total number of hits
il2.misses                    9129 # total number of misses
il2.replacements              8887 # total number of replacements
il2.writebacks                   0 # total number of writebacks
```

$$miss\ rate_{L1} = \frac{114804}{362765} = 0.316$$

$$miss\ rate_{L2} = \frac{9129}{114804} = 0.079$$

Example of result analysis

- Access lower memory level

```
sim: ** simulation statistics **
sim_num_insn                362765 # total number of instructions executed
sim_num_refs                104761 # total number of loads and stores executed
sim_elapsed_time                 1 # total simulation time in seconds
sim_inst_rate           362765.0000 # simulation speed (in insts/sec)
il1.accesses                362765 # total number of accesses
il1.hits                    247961 # total number of hits
il1.misses                  114804 # total number of misses
il1.replacements            114772 # total number of replacements
il1.writebacks                   0 # total number of writebacks
il1.invalidations                0 # total number of invalidations
il1.miss_rate               0.3165 # miss rate (i.e., misses/ref)
il1.repl_rate               0.3164 # replacement rate (i.e., repls/ref)
il1.wb_rate                 0.0000 # writeback rate (i.e., wrbks/ref)
il1.inv_rate                0.0000 # invalidation rate (i.e., invs/ref)
il2.accesses                114804 # total number of accesses
il2.hits                    105675 # total number of hits
il2.misses                    9129 # total number of misses
il2.replacements              8887 # total number of replacements
il2.writebacks                   0 # total number of writebacks
il2.invalidations                0 # total number of invalidations
il2.miss_rate               0.0795 # miss rate (i.e., misses/ref)
il2.repl_rate               0.0774 # replacement rate (i.e., repls/ref)
```

If L1 miss, L2 access

If L2 miss, Main memory access

Unified vs splits

- Unified cache
  - ✓ Data and instructions are stored together
    - Von Neuman architecture
  - ✓ Instructions and data that might map into the same storage locations in a unified cache
  - ✓ Use following, when you make configuration file
    - Ex)

```
-cache:il1 dl1
-cache:dl1 ul1:64:16:1:l

-cache:il2 none          #Instruction L2 cache
-cache:dl2 none          #Data L2 cache
-tlb:itlb none           #Instruction TLB
-tlb:dtlb none           #Data TLB
```

- Split cache
  - ✓ Data and instructions are stored in separate data and instruction caches
    - Harvard architecture
  - ✓ Allows the processor to fetch instructions from instruction cache and data from data cache simultaneously

Simulation 1. Unified vs splits

- Block size = 16, Associativity = 1

- Split cache
  - ✓ Each of instruction cache, Data cache : Number of sets / 2

- Simulate the performance of the cache under the following condition

  - ✓ ./sim-cache –config <config file path> <application path>

# of Sets	Unified Cache Miss rate	Unified Cache AMAT	Split Cache		Split Cache AMAT
			Inst. Miss rate	Data Miss rate	
64					
128					
256					
512					

Simulation 2. L1/L2 size

- Block size = 16, Associativity = 1

  - ✓ Use sim-cache to simulate the performance of the cache under the following condition
  - ✓ Do calculate, once each for instruction-miss rate, data miss rate, inst & data unified cache and AMAT

L1/L1D/L2U	Inst. Miss rate	Data Miss rate	Unified Cache Miss rate	AMAT
8 / 8 / 1024				
16 / 16 / 512				
32 / 32 / 256				
64 / 64 / 128				
128 / 128 / 0				

Simulation 3. Associativity

- Block size = 16
  - ✓ Use sim-cache to simulate the performance of the cache under the following condition

# of Sets	Split Cache Miss rate / AMAT			
	1-way	2-way	3-way	4-way
64				
128				
256				
512				
1024				
2048				

Simulation 4. Block size

- Number of Sets = 512, Associativity = 1
  - ✓ Use sim-cache to simulate the performance of the cache under the following condition

Block size	Unified Cache Miss rate	AMAT
16		
64		
128		
256		
512		

Performance

- Assumption
  - ✓ L1 cache access time is 1 cycle, L2 cache access time is 20 cycles, and Main Memory access time is 200 cycles
  - ✓ If increase twice cache size, cycle time increase 4%
  - ✓ If increase twice associativity, cycle time increase 2%
- Find the cache configuration suitable for SPEC95 program, provided program using AMAT

END