

ECE 466 Project 1

Name: Yinan Fu; UIN: 656721599; NetID: yfu24

1. Performance with default system setup

a. Command

```
sim-outorder -fastfwd 200000000 -max:inst 3000000000 equake.ss<equake.in
```

b. System setup

```
sim: simulation started @ Thu Mar 3 19:59:53 2016, options follow:
# -config                # load configuration from a file
# -dumpconfig            # dump configuration to a file
# -h                     false # print help message
# -v                     false # verbose operation
# -d                     false # enable debug message
# -i                     false # start in Dlite debugger
-seed                    1 # random number generator seed (0 for timer seed)
# -q                     false # initialize and terminate immediately
# -chkpt                 <null> # restore EIO trace execution from <fname>
# -redir:sim             <null> # redirect simulator output to file (non-interactive only)
# -redir:prog            <null> # redirect simulated program output to file
-nice                    0 # simulator scheduling priority
-max:inst                3000000000 # maximum number of inst's to execute
-fastfwd                2000000000 # number of insts skipped before timing starts
# -ptrace               <null> # generate pipetrace, i.e., <fname>|stdout|stderr> <range>
-fetch:ifqsize           4 # instruction fetch queue size (in insts)
-fetch:mplat             3 # extra branch mis-prediction latency
-fetch:speed             1 # speed of front-end of machine relative to execution core
-bpred                  bimod # branch predictor type {nottaken|taken|perfect|bimod|2lev|comb}
-bpred:bimod             2048 # bimodal predictor config (<table size>)
-bpred:2lev              1 1024 8 0 # 2-level predictor config (<l1size> <l2size> <hist_size> <xor>)
-bpred:comb              1024 # combining predictor config (<meta_table_size>)
-bpred:ras               8 # return address stack size (0 for no return stack)
-bpred:btb               512 4 # BTB config (<num_sets> <associativity>)
# -bpred:spec_update     <null> # speculative predictors update in {ID|WB} (default non-spec)
-decode:width            4 # instruction decode B/W (insts/cycle)
-issue:width             4 # instruction issue B/W (insts/cycle)
-issue:inorder           false # run pipeline with in-order issue
-issue:wrongpath         true # issue instructions down wrong execution paths
-commit:width            4 # instruction commit B/W (insts/cycle)
-ruu:size                16 # register update unit (RUU) size
-lsq:size                8 # load/store queue (LSQ) size
-cache:dl1               dl1:128:32:4:l # l1 data cache config, i.e., {<config>|none}
-cache:dl1lat            1 # l1 data cache hit latency (in cycles)
-cache:dl2               ul2:1024:64:4:l # l2 data cache config, i.e., {<config>|none}
-cache:dl2lat            6 # l2 data cache hit latency (in cycles)
-cache:il1               il1:512:32:1:l # l1 inst cache config, i.e., {<config>|dl1|dl2|none}
-cache:il1lat            1 # l1 instruction cache hit latency (in cycles)
-cache:il2               dl2 # l2 instruction cache config, i.e., {<config>|dl2|none}
-cache:il2lat            6 # l2 instruction cache hit latency (in cycles)
-cache:flush             false # flush caches on system calls
-cache:icompress         false # convert 64-bit inst addresses to 32-bit inst equivalents
-mem:lat                 18 2 # memory access latency (<first_chunk> <inter_chunk>)
-mem:width               8 # memory access bus width (in bytes)
-tlb:itlb                itlb:16:4096:4:l # instruction TLB config, i.e., {<config>|none}
-tlb:dtlb                dtlb:32:4096:4:l # data TLB config, i.e., {<config>|none}
-tlb:lat                 30 # inst/data TLB miss latency (in cycles)
-res:ialu                 4 # total number of integer ALU's available
-res:imult                1 # total number of integer multiplier/dividers available
-res:memport             2 # total number of memory system ports available (to CPU)
-res:fpalu                4 # total number of floating point ALU's available
-res:fpmult              1 # total number of floating point multiplier/dividers available
# -pcstat                <null> # profile stat(s) against text addr's (mult uses ok)
-bugcompat               false # operate in backward-compatible bugs mode (for testing only)
```

c. Program output

```
sim: ** fast forwarding 200000000 insts **
equake00: Reading nodes.
equake00: Reading elements.
sim: ** starting performance simulation **
```

d. Performance

sim: ** simulation statistics **

```

sim_num_insn      300000001 # total number of instructions committed
sim_num_refs      98046368 # total number of loads and stores committed
sim_num_loads      68339475 # total number of loads committed
sim_num_stores     29706893.0000 # total number of stores committed
sim_num_branches   78914237 # total number of branches committed
sim_elapsed_time    215 # total simulation time in seconds
sim_inst_rate      1395348.8419 # simulation speed (in insts/sec)
sim_total_insn     316921197 # total number of instructions executed
sim_total_refs     103629216 # total number of loads and stores executed
sim_total_loads     72570922 # total number of loads executed
sim_total_stores    31058294.0000 # total number of stores executed
sim_total_branches  82604306 # total number of branches executed
sim_cycle          179132889 # total simulation time in cycles
sim_IPC            1.6747 # instructions per cycle
sim_CPI            0.5971 # cycles per instruction
sim_exec_BW        1.7692 # total instructions (mis-spec + committed) per cycle
sim_IPB            3.8016 # instruction per branch
IFQ_count          454405113 # cumulative IFQ occupancy
IFQ_fcount         95642807 # cumulative IFQ full count
ifq_occupancy      2.5367 # avg IFQ occupancy (insn's)
ifq_rate           1.7692 # avg IFQ dispatch rate (insn/cycle)
ifq_latency        1.4338 # avg IFQ occupant latency (cycle's)
ifq_full           0.5339 # fraction of time (cycle's) IFQ was full
RUU_count          1818999479 # cumulative RUU occupancy
RUU_fcount         48902994 # cumulative RUU full count
ruu_occupancy      10.1545 # avg RUU occupancy (insn's)
ruu_rate           1.7692 # avg RUU dispatch rate (insn/cycle)
ruu_latency        5.7396 # avg RUU occupant latency (cycle's)
ruu_full           0.2730 # fraction of time (cycle's) RUU was full
LSQ_count          588327218 # cumulative LSQ occupancy
LSQ_fcount         18325000 # cumulative LSQ full count
lsq_occupancy      3.2843 # avg LSQ occupancy (insn's)
lsq_rate           1.7692 # avg LSQ dispatch rate (insn/cycle)
lsq_latency        1.8564 # avg LSQ occupant latency (cycle's)
lsq_full           0.1023 # fraction of time (cycle's) LSQ was full
sim_slip           2715290126 # total number of slip cycles
avg_sim_slip       9.0510 # the average slip between issue and retirement
bpred_bimod.lookups 83954320 # total number of bpred lookups
bpred_bimod.updates 78914235 # total number of updates
bpred_bimod.addr_hits 77473446 # total number of address-predicted hits
bpred_bimod.dir_hits 77473516 # total number of direction-predicted hits (includes addr-hits)
bpred_bimod.misses  1440719 # total number of misses
bpred_bimod.jr_hits 3599818 # total number of address-predicted hits for JR's
bpred_bimod.jr_seen 3599824 # total number of JR's seen
bpred_bimod.jr_non_ras_hits.PP 449569 # total number of address-predicted hits for non-RAS JR's
bpred_bimod.jr_non_ras_seen.PP 449570 # total number of non-RAS JR's seen
bpred_bimod.bpred_addr_rate 0.9817 # branch address-prediction rate (i.e., addr-hits/updates)
bpred_bimod.bpred_dir_rate 0.9817 # branch direction-prediction rate (i.e., all-hits/updates)
bpred_bimod.bpred_jr_rate 1.0000 # JR address-prediction rate (i.e., JR addr-hits/JRs seen)
bpred_bimod.bpred_jr_non_ras_rate.PP 1.0000 # non-RAS JR addr-pred rate (ie, non-RAS JR hits/JRs seen)
bpred_bimod.retstack_pushes 3241508 # total number of address pushed onto ret-addr stack
bpred_bimod.retstack_pops 3150293 # total number of address popped off of ret-addr stack
bpred_bimod.used_ras.PP 3150254 # total number of RAS predictions used
bpred_bimod.ras_hits.PP 3150249 # total number of RAS hits
bpred_bimod.ras_rate.PP 1.0000 # RAS prediction rate (i.e., RAS hits/used RAS)
i11.accesses       329973081 # total number of accesses
i11.hits           322322360 # total number of hits
i11.misses         7650721 # total number of misses
i11.replacements   7650506 # total number of replacements
i11.writebacks      0 # total number of writebacks
i11.invalidations   0 # total number of invalidations
i11.miss_rate       0.0232 # miss rate (i.e., misses/ref)
i11.repl_rate       0.0232 # replacement rate (i.e., repls/ref)
i11.wb_rate         0.0000 # writeback rate (i.e., wrbks/ref)
i11.inv_rate        0.0000 # invalidation rate (i.e., invs/ref)
d11.accesses       99036937 # total number of accesses
d11.hits           98967530 # total number of hits
d11.misses         69407 # total number of misses
d11.replacements   68895 # total number of replacements
d11.writebacks      56171 # total number of writebacks
d11.invalidations   0 # total number of invalidations
d11.miss_rate       0.0007 # miss rate (i.e., misses/ref)
d11.repl_rate       0.0007 # replacement rate (i.e., repls/ref)
d11.wb_rate         0.0006 # writeback rate (i.e., wrbks/ref)
d11.inv_rate        0.0000 # invalidation rate (i.e., invs/ref)

```

ul2.accesses	7776299 # total number of accesses
ul2.hits	7742316 # total number of hits
ul2.misses	33983 # total number of misses
ul2.replacements	29887 # total number of replacements
ul2.writebacks	24574 # total number of writebacks
ul2.invalidations	0 # total number of invalidations
ul2.miss_rate	0.0044 # miss rate (i.e., misses/ref)
ul2.repl_rate	0.0038 # replacement rate (i.e., repls/ref)
ul2.wb_rate	0.0032 # writeback rate (i.e., wrbks/ref)
ul2.inv_rate	0.0000 # invalidation rate (i.e., invs/ref)
itlb.accesses	329973081 # total number of accesses
itlb.hits	329973069 # total number of hits
itlb.misses	12 # total number of misses
itlb.replacements	0 # total number of replacements
itlb.writebacks	0 # total number of writebacks
itlb.invalidations	0 # total number of invalidations
itlb.miss_rate	0.0000 # miss rate (i.e., misses/ref)
itlb.repl_rate	0.0000 # replacement rate (i.e., repls/ref)
itlb.wb_rate	0.0000 # writeback rate (i.e., wrbks/ref)
itlb.inv_rate	0.0000 # invalidation rate (i.e., invs/ref)
dtlb.accesses	100835273 # total number of accesses
dtlb.hits	100834736 # total number of hits
dtlb.misses	537 # total number of misses
dtlb.replacements	409 # total number of replacements
dtlb.writebacks	0 # total number of writebacks
dtlb.invalidations	0 # total number of invalidations
dtlb.miss_rate	0.0000 # miss rate (i.e., misses/ref)
dtlb.repl_rate	0.0000 # replacement rate (i.e., repls/ref)
dtlb.wb_rate	0.0000 # writeback rate (i.e., wrbks/ref)
dtlb.inv_rate	0.0000 # invalidation rate (i.e., invs/ref)
sim_invalid_addrs	0 # total non-speculative bogus addresses seen (debug var)
ld_text_base	0x00400000 # program text (code) segment base
ld_text_size	132784 # program text (code) size in bytes
ld_data_base	0x10000000 # program initialized data segment base
ld_data_size	16384 # program init'ed '.data' and uninit'ed '.bss' size in bytes
ld_stack_base	0x7fff0000 # program stack segment base (highest address in stack)
ld_stack_size	16384 # program initial stack size
ld_prog_entry	0x00400140 # program entry point (initial PC)
ld_environ_base	0x7fff8000 # program environment base address address
ld_target_big_endian	0 # target executable endian-ness, non-zero if big endian
mem.page_count	1176 # total number of pages allocated
mem.page_mem	4704k # total size of memory pages allocated
mem.ptab_misses	6280 # total first level page table misses
mem.ptab_accesses	3321066400 # total page table accesses
mem.ptab_miss_rate	0.0000 # first level page table miss rate

Comments:

The IPC and CPI values for this simulation are recorded by “ $\begin{cases} \text{sim_IPC} & 1.6747 \text{ \# instructions per cycle,} \\ \text{sim_CPI} & 0.5971 \text{ \# cycles per instruction} \end{cases}$ ”

And the total number of committed instructions is 300000001. Total number of cycles is 179132889. Thus, from these two numbers, we can compute:

$$IPC = \frac{300000001}{179132889} \approx 1.6747 \left(\frac{\text{insts}}{\text{cycle}} \right); CPI = \frac{1}{IPC} \approx 0.5971 \left(\frac{\text{cycles}}{\text{inst}} \right), \text{ which are exactly the same}$$

as *sim_IPC* and *sim_CPI*.

2. Performance with in-order execution

a. Command

```
sim-outorder -fastfwd 200000000 -max:inst 300000000 -issue:inorder true equake.ss<equake.in
```

b. System setup

sim: simulation started @ Thu Mar 3 20:07:11 2016, options follow:

```
# -config          # load configuration from a file
# -dumpconfig     # dump configuration to a file
# -h              false # print help message
# -v              false # verbose operation
```

```

# -d                false # enable debug message
# -i                false # start in Dlite debugger
-seed                1 # random number generator seed (0 for timer seed)
# -q                false # initialize and terminate immediately
# -chkpt            <null> # restore EIO trace execution from <fname>
# -redir:sim        <null> # redirect simulator output to file (non-interactive only)
# -redir:prog       <null> # redirect simulated program output to file
-nice                0 # simulator scheduling priority
-max:inst            300000000 # maximum number of inst's to execute
-fastfwd            200000000 # number of insts skipped before timing starts
# -ptrace           <null> # generate pipetrace, i.e., <fname>|stdout|stderr> <range>
-fetch:ifqsize      4 # instruction fetch queue size (in insts)
-fetch:mplat        3 # extra branch mis-prediction latency
-fetch:speed        1 # speed of front-end of machine relative to execution core
-bpred              bimod # branch predictor type {nottaken|taken|perfect|bimod|2lev|comb}
-bpred:bimod        2048 # bimodal predictor config (<table size>)
-bpred:2lev         1 1024 8 0 # 2-level predictor config (<l1size> <l2size> <hist_size> <xor>)
-bpred:comb         1024 # combining predictor config (<meta_table_size>)
-bpred:ras          8 # return address stack size (0 for no return stack)
-bpred:btb          512 4 # BTB config (<num_sets> <associativity>)
# -bpred:spec_update <null> # speculative predictors update in {ID|WB} (default non-spec)
-decode:width       4 # instruction decode B/W (insts/cycle)
-issue:width        4 # instruction issue B/W (insts/cycle)
-issue:inorder      true # run pipeline with in-order issue
-issue:wrongpath    true # issue instructions down wrong execution paths
-commit:width       4 # instruction commit B/W (insts/cycle)
-ruu:size           16 # register update unit (RUU) size
-lsq:size           8 # load/store queue (LSQ) size
-cache:dl1           dl1:128:32:4:l # l1 data cache config, i.e., {<config>|none}
-cache:dl1lat        1 # l1 data cache hit latency (in cycles)
-cache:dl2           ul2:1024:64:4:l # l2 data cache config, i.e., {<config>|none}
-cache:dl2lat        6 # l2 data cache hit latency (in cycles)
-cache:il1           il1:512:32:1:l # l1 inst cache config, i.e., {<config>|dl1|dl2|none}
-cache:il1lat        1 # l1 instruction cache hit latency (in cycles)
-cache:il2           dl2 # l2 instruction cache config, i.e., {<config>|dl2|none}
-cache:il2lat        6 # l2 instruction cache hit latency (in cycles)
-cache:flush        false # flush caches on system calls
-cache:icompress    false # convert 64-bit inst addresses to 32-bit inst equivalents
-mem:lat            18 2 # memory access latency (<first_chunk> <inter_chunk>)
-mem:width          8 # memory access bus width (in bytes)
-tlb:itlb           itlb:16:4096:4:l # instruction TLB config, i.e., {<config>|none}
-tlb:dtlb           dtlb:32:4096:4:l # data TLB config, i.e., {<config>|none}
-tlb:lat            30 # inst/data TLB miss latency (in cycles)
-res:ialu           4 # total number of integer ALU's available
-res:imult          1 # total number of integer multiplier/dividers available
-res:memport        2 # total number of memory system ports available (to CPU)
-res:fpalu          4 # total number of floating point ALU's available
-res:fpmult         1 # total number of floating point multiplier/dividers available
# -pcstat           <null> # profile stat(s) against text addr's (mult uses ok)
-bugcompat          false # operate in backward-compatible bugs mode (for testing only)

```

c. Program output

```

sim: ** fast forwarding 200000000 insts **
equake00: Reading nodes.
equake00: Reading elements.
sim: ** starting performance simulation **

```

d. Performance

```

sim: ** simulation statistics **
sim_num_insn        300000000 # total number of instructions committed
sim_num_refs        98046367 # total number of loads and stores committed
sim_num_loads        68339474 # total number of loads committed
sim_num_stores       29706893.0000 # total number of stores committed
sim_num_branches     78914237 # total number of branches committed
sim_elapsed_time     197 # total simulation time in seconds
sim_inst_rate        1522842.6396 # simulation speed (in insts/sec)
sim_total_insn       302340591 # total number of instructions executed
sim_total_refs       98587898 # total number of loads and stores executed
sim_total_loads       68880312 # total number of loads executed
sim_total_stores      29707586.0000 # total number of stores executed
sim_total_branches    78914326 # total number of branches executed
sim_cycle            386814232 # total simulation time in cycles
sim_ipc              0.7756 # instructions per cycle
sim_cpi              1.2894 # cycles per instruction
sim_exec_BW          0.7816 # total instructions (mis-spec + committed) per cycle

```

sim_IPB	3.8016 # instruction per branch
IFQ_count	1341116038 # cumulative IFQ occupancy
IFQ_fcount	316774185 # cumulative IFQ full count
ifq_occupancy	3.4671 # avg IFQ occupancy (insn's)
ifq_rate	0.7816 # avg IFQ dispatch rate (insn/cycle)
ifq_latency	4.4358 # avg IFQ occupant latency (cycle's)
ifq_full	0.8189 # fraction of time (cycle's) IFQ was full
RUU_count	1045107571 # cumulative RUU occupancy
RUU_fcount	0 # cumulative RUU full count
ruu_occupancy	2.7018 # avg RUU occupancy (insn's)
ruu_rate	0.7816 # avg RUU dispatch rate (insn/cycle)
ruu_latency	3.4567 # avg RUU occupant latency (cycle's)
ruu_full	0.0000 # fraction of time (cycle's) RUU was full
LSQ_count	367080264 # cumulative LSQ occupancy
LSQ_fcount	0 # cumulative LSQ full count
lsq_occupancy	0.9490 # avg LSQ occupancy (insn's)
lsq_rate	0.7816 # avg LSQ dispatch rate (insn/cycle)
lsq_latency	1.2141 # avg LSQ occupant latency (cycle's)
lsq_full	0.0000 # fraction of time (cycle's) LSQ was full
sim_slip	1807351389 # total number of slip cycles
avg_sim_slip	6.0245 # the average slip between issue and retirement
bpred_bimod.lookups	80265119 # total number of bpred lookups
bpred_bimod.updates	78914237 # total number of updates
bpred_bimod.addr_hits	77473448 # total number of address-predicted hits
bpred_bimod.dir_hits	77473518 # total number of direction-predicted hits (includes addr-hits)
bpred_bimod.misses	1440719 # total number of misses
bpred_bimod.jr_hits	3599818 # total number of address-predicted hits for JR's
bpred_bimod.jr_seen	3599824 # total number of JR's seen
bpred_bimod.jr_non_ras_hits.PP	449569 # total number of address-predicted hits for non-RAS JR's
bpred_bimod.jr_non_ras_seen.PP	449570 # total number of non-RAS JR's seen
bpred_bimod.bpred_addr_rate	0.9817 # branch address-prediction rate (i.e., addr-hits/updates)
bpred_bimod.bpred_dir_rate	0.9817 # branch direction-prediction rate (i.e., all-hits/updates)
bpred_bimod.bpred_jr_rate	1.0000 # JR address-prediction rate (i.e., JR addr-hits/JRs seen)
bpred_bimod.bpred_jr_non_ras_rate.PP	1.0000 # non-RAS JR addr-pred rate (ie, non-RAS JR hits/JRs seen)
bpred_bimod.retstack_pushes	3150906 # total number of address pushed onto ret-addr stack
bpred_bimod.retstack_pops	3150254 # total number of address popped off of ret-addr stack
bpred_bimod.used_ras.PP	3150254 # total number of RAS predictions used
bpred_bimod.ras_hits.PP	3150249 # total number of RAS hits
bpred_bimod.ras_rate.PP	1.0000 # RAS prediction rate (i.e., RAS hits/used RAS)
il1.accesses	315753897 # total number of accesses
il1.hits	308103178 # total number of hits
il1.misses	7650719 # total number of misses
il1.replacements	7650506 # total number of replacements
il1.writebacks	0 # total number of writebacks
il1.invalidations	0 # total number of invalidations
il1.miss_rate	0.0242 # miss rate (i.e., misses/ref)
il1.repl_rate	0.0242 # 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)
dl1.accesses	98046366 # total number of accesses
dl1.hits	97977091 # total number of hits
dl1.misses	69275 # total number of misses
dl1.replacements	68763 # total number of replacements
dl1.writebacks	56139 # total number of writebacks
dl1.invalidations	0 # total number of invalidations
dl1.miss_rate	0.0007 # miss rate (i.e., misses/ref)
dl1.repl_rate	0.0007 # replacement rate (i.e., repls/ref)
dl1.wb_rate	0.0006 # writeback rate (i.e., wrbks/ref)
dl1.inv_rate	0.0000 # invalidation rate (i.e., invs/ref)
ul2.accesses	7776133 # total number of accesses
ul2.hits	7742152 # total number of hits
ul2.misses	33981 # total number of misses
ul2.replacements	29885 # total number of replacements
ul2.writebacks	24573 # total number of writebacks
ul2.invalidations	0 # total number of invalidations
ul2.miss_rate	0.0044 # miss rate (i.e., misses/ref)
ul2.repl_rate	0.0038 # replacement rate (i.e., repls/ref)
ul2.wb_rate	0.0032 # writeback rate (i.e., wrbks/ref)
ul2.inv_rate	0.0000 # invalidation rate (i.e., invs/ref)
itlb.accesses	315753897 # total number of accesses
itlb.hits	315753885 # total number of hits
itlb.misses	12 # total number of misses
itlb.replacements	0 # total number of replacements
itlb.writebacks	0 # total number of writebacks
itlb.invalidations	0 # total number of invalidations
itlb.miss_rate	0.0000 # miss rate (i.e., misses/ref)
itlb.repl_rate	0.0000 # replacement rate (i.e., repls/ref)
itlb.wb_rate	0.0000 # writeback rate (i.e., wrbks/ref)

itlb.inv_rate	0.0000 # invalidation rate (i.e., invs/ref)
dtlb.accesses	98046366 # total number of accesses
dtlb.hits	98045830 # total number of hits
dtlb.misses	536 # total number of misses
dtlb.replacements	408 # total number of replacements
dtlb.writebacks	0 # total number of writebacks
dtlb.invalidations	0 # total number of invalidations
dtlb.miss_rate	0.0000 # miss rate (i.e., misses/ref)
dtlb.repl_rate	0.0000 # replacement rate (i.e., repls/ref)
dtlb.wb_rate	0.0000 # writeback rate (i.e., wrbks/ref)
dtlb.inv_rate	0.0000 # invalidation rate (i.e., invs/ref)
sim_invalid_addr	0 # total non-speculative bogus addresses seen (debug var)
ld_text_base	0x00400000 # program text (code) segment base
ld_text_size	132784 # program text (code) size in bytes
ld_data_base	0x10000000 # program initialized data segment base
ld_data_size	16384 # program init'ed '.data' and uninit'ed '.bss' size in bytes
ld_stack_base	0x7fffc000 # program stack segment base (highest address in stack)
ld_stack_size	16384 # program initial stack size
ld_prog_entry	0x00400140 # program entry point (initial PC)
ld_environ_base	0x7fff8000 # program environment base address address
ld_target_big_endian	0 # target executable endianness, non-zero if big endian
mem.page_count	1176 # total number of pages allocated
mem.page_mem	4704k # total size of memory pages allocated
mem.ptab_misses	6280 # total first level page table misses
mem.ptab_accesses	3256809746 # total page table accesses
mem.ptab_miss_rate	0.0000 # first level page table miss rate

Comments:

This time, the total simulation time in cycles increases to 386814232. And the total number of committed instruction only decrease by 1, which is 300000000. The IPC and CPI values for this simulation are recorded by “ $\begin{cases} \text{sim_IPC} & 0.7756 \text{ \# instructions per cycle,} \\ \text{sim_CPI} & 1.2894 \text{ \# cycles per instruction} \end{cases}$ ”. Note that this simulation just focuses on one

task running at PISA machine. Hence, the performance should use IPC value or CPI value as criteria. Hence,

the performance of out-of-order execution is $\text{Speedup} = \frac{IPC_{out-of-order}}{IPC_{in-order}} = \frac{1.6747}{0.7756} \approx 2.1592$ times better

than in-order execution. Performance loss is $\frac{IPC_{out-of-order} - IPC_{in-order}}{IPC_{out-of-order}} = \frac{1.6747 - 0.7756}{1.6747} \approx 53.69\%$

3. Running time estimation

In order to estimate how fast this program will run on difference devices, we need to find out how many instructions this program contains by using “sim-safe equake.ss<equake.in”. The results are printed as following.

sim-safe: SimpleScalar/PISA Tool Set version 3.0 of August, 2003.
Copyright (c) 1994-2003 by Todd M. Austin, Ph.D. and SimpleScalar, LLC.
All Rights Reserved. This version of SimpleScalar is licensed for academic non-commercial use. No portion of this work may be used by any commercial entity, or for any commercial purpose, without the prior written permission of SimpleScalar, LLC (info@simplescalar.com).

sim: command line: sim-safe equake.ss

sim: simulation started @ Thu Mar 3 20:25:51 2016, options follow:

sim-safe: This simulator implements a functional simulator. This functional simulator is the simplest, most user-friendly simulator in the simplescalar tool set. Unlike sim-fast, this functional simulator checks for all instruction errors, and the implementation is crafted for clarity rather than speed.

```
# -config          # load configuration from a file
# -dumpconfig     # dump configuration to a file
# -h              false # print help message
# -v              false # verbose operation
# -d              false # enable debug message
# -i              false # start in Dlite debugger
-seed             1 # random number generator seed (0 for timer seed)
```

```
# -q                false # initialize and terminate immediately
# -chkpt           <null> # restore EIO trace execution from <fname>
# -redir:sim       <null> # redirect simulator output to file (non-interactive only)
# -redir:prog      <null> # redirect simulated program output to file
# -nice            0 # simulator scheduling priority
# -max:inst        0 # maximum number of inst's to execute
```

```
sim: ** starting functional simulation **
equake00: Reading nodes.
equake00: Reading elements.
equake00: Reading sparse matrix structure.
equake00: Beginning simulation.
```

CASE SUMMARY

Fault information

```
Orientation: strike: 1.937315
              dip: 0.767945
              rake: 1.221730
              dislocation: 29.640788 cm
```

```
Hypocenter: (32.264153, 23.814432, -11.250000) Km
```

Excitation characteristics

```
Time step: 0.002400 sec
Duration: 9.250000 sec
Rise time: 0.600000 sec
```

```
The source is node 5903 at (32.250000 24.364300 -11.200000)
The epicenter is node 16745 at (32.250000 23.687500 0.000000)
```

Time step 30

```
5903: -3.42e-02 -3.11e-02 -4.03e-02
16745: 9.04e-35 -8.75e-34 4.41e-33
```

Time step 60

```
5903: -2.46e-01 -2.27e-01 -2.98e-01
16745: 2.45e-27 -2.59e-26 1.30e-25
```

Time step 90

```
5903: -7.38e-01 -6.95e-01 -9.22e-01
16745: 4.35e-23 -5.32e-22 2.68e-21
```

Time step 120

```
5903: -1.49e+00 -1.44e+00 -1.92e+00
16745: 3.74e-20 -5.61e-19 2.83e-18
```

Time step 150

```
5903: -2.37e+00 -2.34e+00 -3.12e+00
16745: 5.89e-18 -1.14e-16 5.76e-16
```

Time step 180

```
5903: -3.17e+00 -3.21e+00 -4.26e+00
16745: 3.21e-16 -8.11e-15 4.08e-14
```

Time step 210

```
5903: -3.73e+00 -3.88e+00 -5.10e+00
16745: 9.19e-15 -2.76e-13 1.38e-12
```

Time step 240

```
5903: -4.00e+00 -4.27e+00 -5.57e+00
16745: 1.91e-13 -5.42e-12 2.70e-11
```

Time step 270

```
5903: -4.08e+00 -4.46e+00 -5.78e+00
16745: 3.28e-12 -7.00e-11 3.45e-10
```

Time step 300

```
5903: -4.06e+00 -4.52e+00 -5.86e+00
16745: 4.41e-11 -6.45e-10 3.14e-09
```

Time step 330

```
5903: -3.98e+00 -4.51e+00 -5.90e+00
16745: 4.49e-10 -4.50e-09 2.18e-08
```

Time step 360

```
5903: -3.91e+00 -4.49e+00 -5.96e+00
16745: 3.47e-09 -2.49e-08 1.20e-07
```

Time step 390

```
5903: -3.86e+00 -4.48e+00 -6.07e+00
16745: 2.10e-08 -1.13e-07 5.42e-07
```

Time step 420

```
5903: -3.85e+00 -4.48e+00 -6.21e+00
16745: 1.02e-07 -4.35e-07 2.08e-06
```

Time step 450

```
5903: -3.86e+00 -4.49e+00 -6.36e+00
16745: 4.12e-07 -1.44e-06 6.89e-06
```

Time step 480

```
5903: -3.89e+00 -4.50e+00 -6.48e+00
16745: 1.40e-06 -4.17e-06 2.01e-05
```

Time step 510

```
5903: -3.91e+00 -4.50e+00 -6.57e+00
```

16745: 4.12e-06 -1.08e-05 5.26e-05
Time step 540
5903: -3.93e+00 -4.50e+00 -6.62e+00
16745: 1.06e-05 -2.53e-05 1.24e-04
Time step 570
5903: -3.95e+00 -4.50e+00 -6.65e+00
16745: 2.40e-05 -5.46e-05 2.69e-04
Time step 600
5903: -3.96e+00 -4.51e+00 -6.67e+00
16745: 4.85e-05 -1.09e-04 5.38e-04
Time step 630
5903: -3.97e+00 -4.52e+00 -6.67e+00
16745: 8.80e-05 -2.06e-04 1.00e-03
Time step 660
5903: -3.98e+00 -4.54e+00 -6.68e+00
16745: 1.43e-04 -3.68e-04 1.74e-03
Time step 690
5903: -3.98e+00 -4.56e+00 -6.67e+00
16745: 2.09e-04 -6.26e-04 2.84e-03
Time step 720
5903: -3.98e+00 -4.57e+00 -6.67e+00
16745: 2.70e-04 -1.02e-03 4.36e-03
Time step 750
5903: -3.98e+00 -4.59e+00 -6.66e+00
16745: 3.01e-04 -1.60e-03 6.31e-03
Time step 780
5903: -3.98e+00 -4.60e+00 -6.65e+00
16745: 2.64e-04 -2.41e-03 8.59e-03
Time step 810
5903: -3.97e+00 -4.60e+00 -6.63e+00
16745: 1.20e-04 -3.46e-03 1.10e-02
Time step 840
5903: -3.97e+00 -4.60e+00 -6.62e+00
16745: -1.70e-04 -4.77e-03 1.31e-02
Time step 870
5903: -3.97e+00 -4.61e+00 -6.61e+00
16745: -6.32e-04 -6.28e-03 1.44e-02
Time step 900
5903: -3.98e+00 -4.60e+00 -6.62e+00
16745: -1.27e-03 -7.90e-03 1.43e-02
Time step 930
5903: -3.98e+00 -4.60e+00 -6.63e+00
16745: -2.07e-03 -9.50e-03 1.22e-02
Time step 960
5903: -3.99e+00 -4.60e+00 -6.64e+00
16745: -2.95e-03 -1.09e-02 7.97e-03
Time step 990
5903: -3.99e+00 -4.60e+00 -6.66e+00
16745: -3.81e-03 -1.20e-02 1.42e-03
Time step 1020
5903: -4.00e+00 -4.60e+00 -6.69e+00
16745: -4.52e-03 -1.26e-02 -7.06e-03
Time step 1050
5903: -4.00e+00 -4.60e+00 -6.70e+00
16745: -4.90e-03 -1.26e-02 -1.68e-02
Time step 1080
5903: -4.00e+00 -4.61e+00 -6.71e+00
16745: -4.81e-03 -1.19e-02 -2.69e-02
Time step 1110
5903: -4.00e+00 -4.61e+00 -6.71e+00
16745: -4.11e-03 -1.04e-02 -3.65e-02
Time step 1140
5903: -3.99e+00 -4.61e+00 -6.70e+00
16745: -2.80e-03 -8.05e-03 -4.47e-02
Time step 1170
5903: -3.98e+00 -4.61e+00 -6.70e+00
16745: -9.48e-04 -4.85e-03 -5.13e-02
Time step 1200
5903: -3.98e+00 -4.61e+00 -6.69e+00
16745: 1.24e-03 -8.07e-04 -5.62e-02
Time step 1230
5903: -3.97e+00 -4.61e+00 -6.69e+00
16745: 3.50e-03 3.90e-03 -5.96e-02
Time step 1260
5903: -3.97e+00 -4.61e+00 -6.69e+00
16745: 5.51e-03 8.96e-03 -6.22e-02
Time step 1290
5903: -3.98e+00 -4.61e+00 -6.70e+00

16745: 7.02e-03 1.39e-02 -6.46e-02
Time step 1320
5903: -3.99e+00 -4.61e+00 -6.71e+00
16745: 7.88e-03 1.84e-02 -6.71e-02
Time step 1350
5903: -3.99e+00 -4.61e+00 -6.71e+00
16745: 8.13e-03 2.19e-02 -7.02e-02
Time step 1380
5903: -4.00e+00 -4.61e+00 -6.72e+00
16745: 8.02e-03 2.45e-02 -7.39e-02
Time step 1410
5903: -4.01e+00 -4.61e+00 -6.72e+00
16745: 7.97e-03 2.61e-02 -7.81e-02
Time step 1440
5903: -4.01e+00 -4.62e+00 -6.71e+00
16745: 8.47e-03 2.74e-02 -8.24e-02
Time step 1470
5903: -4.01e+00 -4.62e+00 -6.71e+00
16745: 9.95e-03 2.88e-02 -8.63e-02
Time step 1500
5903: -4.00e+00 -4.62e+00 -6.70e+00
16745: 1.26e-02 3.08e-02 -8.93e-02
Time step 1530
5903: -4.00e+00 -4.62e+00 -6.70e+00
16745: 1.61e-02 3.38e-02 -9.11e-02
Time step 1560
5903: -3.99e+00 -4.62e+00 -6.71e+00
16745: 2.00e-02 3.78e-02 -9.15e-02
Time step 1590
5903: -3.99e+00 -4.62e+00 -6.71e+00
16745: 2.33e-02 4.27e-02 -9.07e-02
Time step 1620
5903: -3.99e+00 -4.62e+00 -6.72e+00
16745: 2.52e-02 4.78e-02 -8.89e-02
Time step 1650
5903: -3.99e+00 -4.63e+00 -6.72e+00
16745: 2.54e-02 5.25e-02 -8.67e-02
Time step 1680
5903: -3.99e+00 -4.63e+00 -6.72e+00
16745: 2.37e-02 5.59e-02 -8.49e-02
Time step 1710
5903: -3.99e+00 -4.63e+00 -6.72e+00
16745: 2.09e-02 5.72e-02 -8.37e-02
Time step 1740
5903: -3.99e+00 -4.62e+00 -6.72e+00
16745: 1.77e-02 5.58e-02 -8.35e-02
Time step 1770
5903: -3.99e+00 -4.62e+00 -6.71e+00
16745: 1.51e-02 5.16e-02 -8.41e-02
Time step 1800
5903: -3.99e+00 -4.62e+00 -6.70e+00
16745: 1.37e-02 4.48e-02 -8.53e-02
Time step 1830
5903: -3.99e+00 -4.61e+00 -6.70e+00
16745: 1.34e-02 3.64e-02 -8.67e-02
Time step 1860
5903: -3.99e+00 -4.61e+00 -6.70e+00
16745: 1.38e-02 2.75e-02 -8.79e-02
Time step 1890
5903: -3.99e+00 -4.61e+00 -6.71e+00
16745: 1.41e-02 1.93e-02 -8.82e-02
Time step 1920
5903: -3.99e+00 -4.61e+00 -6.72e+00
16745: 1.37e-02 1.28e-02 -8.73e-02
Time step 1950
5903: -3.99e+00 -4.61e+00 -6.72e+00
16745: 1.21e-02 8.32e-03 -8.49e-02
Time step 1980
5903: -4.00e+00 -4.62e+00 -6.73e+00
16745: 9.36e-03 6.00e-03 -8.12e-02
Time step 2010
5903: -4.00e+00 -4.62e+00 -6.74e+00
16745: 5.83e-03 5.39e-03 -7.69e-02
Time step 2040
5903: -4.00e+00 -4.62e+00 -6.74e+00
16745: 2.02e-03 5.86e-03 -7.32e-02
Time step 2070
5903: -3.99e+00 -4.62e+00 -6.75e+00

16745: -1.59e-03 6.73e-03 -7.15e-02
Time step 2100
5903: -3.99e+00 -4.61e+00 -6.74e+00
16745: -4.65e-03 7.35e-03 -7.29e-02
Time step 2130
5903: -3.99e+00 -4.61e+00 -6.74e+00
16745: -6.96e-03 7.32e-03 -7.76e-02
Time step 2160
5903: -3.99e+00 -4.61e+00 -6.74e+00
16745: -8.40e-03 6.54e-03 -8.52e-02
Time step 2190
5903: -3.99e+00 -4.62e+00 -6.74e+00
16745: -8.85e-03 5.26e-03 -9.40e-02
Time step 2220
5903: -3.99e+00 -4.62e+00 -6.73e+00
16745: -8.19e-03 4.00e-03 -1.02e-01
Time step 2250
5903: -3.99e+00 -4.62e+00 -6.73e+00
16745: -6.38e-03 3.39e-03 -1.08e-01
Time step 2280
5903: -3.99e+00 -4.62e+00 -6.74e+00
16745: -3.57e-03 3.94e-03 -1.09e-01
Time step 2310
5903: -3.99e+00 -4.62e+00 -6.74e+00
16745: -1.01e-04 5.90e-03 -1.06e-01
Time step 2340
5903: -3.99e+00 -4.62e+00 -6.74e+00
16745: 3.50e-03 9.25e-03 -9.96e-02
Time step 2370
5903: -3.99e+00 -4.62e+00 -6.74e+00
16745: 6.74e-03 1.38e-02 -9.05e-02
Time step 2400
5903: -3.99e+00 -4.61e+00 -6.74e+00
16745: 9.30e-03 1.92e-02 -8.16e-02
Time step 2430
5903: -3.99e+00 -4.61e+00 -6.74e+00
16745: 1.11e-02 2.51e-02 -7.53e-02
Time step 2460
5903: -3.99e+00 -4.61e+00 -6.74e+00
16745: 1.21e-02 3.10e-02 -7.37e-02
Time step 2490
5903: -3.99e+00 -4.61e+00 -6.75e+00
16745: 1.24e-02 3.57e-02 -7.71e-02
Time step 2520
5903: -3.99e+00 -4.62e+00 -6.75e+00
16745: 1.18e-02 3.79e-02 -8.48e-02
Time step 2550
5903: -3.99e+00 -4.62e+00 -6.76e+00
16745: 1.03e-02 3.64e-02 -9.46e-02
Time step 2580
5903: -3.99e+00 -4.62e+00 -6.76e+00
16745: 7.62e-03 3.03e-02 -1.04e-01
Time step 2610
5903: -3.99e+00 -4.62e+00 -6.76e+00
16745: 4.18e-03 1.99e-02 -1.10e-01
Time step 2640
5903: -3.99e+00 -4.62e+00 -6.76e+00
16745: 5.35e-04 7.14e-03 -1.13e-01
Time step 2670
5903: -3.99e+00 -4.62e+00 -6.76e+00
16745: -2.56e-03 -5.43e-03 -1.12e-01
Time step 2700
5903: -3.99e+00 -4.62e+00 -6.76e+00
16745: -4.49e-03 -1.48e-02 -1.08e-01
Time step 2730
5903: -3.99e+00 -4.62e+00 -6.75e+00
16745: -5.05e-03 -1.89e-02 -1.02e-01
Time step 2760
5903: -3.99e+00 -4.62e+00 -6.75e+00
16745: -4.49e-03 -1.70e-02 -9.61e-02
Time step 2790
5903: -3.99e+00 -4.62e+00 -6.75e+00
16745: -3.29e-03 -1.02e-02 -9.08e-02
Time step 2820
5903: -3.99e+00 -4.62e+00 -6.75e+00
16745: -1.89e-03 -7.66e-04 -8.67e-02
Time step 2850
5903: -3.99e+00 -4.62e+00 -6.75e+00

16745: -4.83e-04 8.74e-03 -8.44e-02
Time step 2880
5903: -3.99e+00 -4.62e+00 -6.75e+00
16745: 9.58e-04 1.62e-02 -8.41e-02
Time step 2910
5903: -3.99e+00 -4.62e+00 -6.75e+00
16745: 2.51e-03 2.08e-02 -8.60e-02
Time step 2940
5903: -3.99e+00 -4.62e+00 -6.76e+00
16745: 4.11e-03 2.26e-02 -9.01e-02
Time step 2970
5903: -3.99e+00 -4.62e+00 -6.76e+00
16745: 5.51e-03 2.27e-02 -9.57e-02
Time step 3000
5903: -3.99e+00 -4.62e+00 -6.76e+00
16745: 6.42e-03 2.22e-02 -1.02e-01
Time step 3030
5903: -3.98e+00 -4.62e+00 -6.76e+00
16745: 6.71e-03 2.20e-02 -1.07e-01
Time step 3060
5903: -3.98e+00 -4.61e+00 -6.76e+00
16745: 6.49e-03 2.23e-02 -1.11e-01
Time step 3090
5903: -3.98e+00 -4.61e+00 -6.76e+00
16745: 6.06e-03 2.26e-02 -1.13e-01
Time step 3120
5903: -3.98e+00 -4.61e+00 -6.76e+00
16745: 5.65e-03 2.23e-02 -1.12e-01
Time step 3150
5903: -3.98e+00 -4.61e+00 -6.76e+00
16745: 5.31e-03 2.07e-02 -1.08e-01
Time step 3180
5903: -3.98e+00 -4.61e+00 -6.76e+00
16745: 4.88e-03 1.77e-02 -1.03e-01
Time step 3210
5903: -3.98e+00 -4.61e+00 -6.76e+00
16745: 4.10e-03 1.38e-02 -9.78e-02
Time step 3240
5903: -3.98e+00 -4.61e+00 -6.76e+00
16745: 2.83e-03 9.74e-03 -9.29e-02
Time step 3270
5903: -3.98e+00 -4.61e+00 -6.76e+00
16745: 1.15e-03 6.89e-03 -8.91e-02
Time step 3300
5903: -3.98e+00 -4.61e+00 -6.76e+00
16745: -6.79e-04 6.27e-03 -8.66e-02
Time step 3330
5903: -3.98e+00 -4.61e+00 -6.76e+00
16745: -2.34e-03 8.46e-03 -8.52e-02
Time step 3360
5903: -3.98e+00 -4.61e+00 -6.76e+00
16745: -3.53e-03 1.32e-02 -8.49e-02
Time step 3390
5903: -3.98e+00 -4.61e+00 -6.76e+00
16745: -4.03e-03 1.96e-02 -8.57e-02
Time step 3420
5903: -3.98e+00 -4.61e+00 -6.76e+00
16745: -3.71e-03 2.58e-02 -8.78e-02
Time step 3450
5903: -3.98e+00 -4.61e+00 -6.76e+00
16745: -2.56e-03 3.00e-02 -9.16e-02
Time step 3480
5903: -3.99e+00 -4.61e+00 -6.76e+00
16745: -7.82e-04 3.07e-02 -9.70e-02
Time step 3510
5903: -3.99e+00 -4.62e+00 -6.76e+00
16745: 1.23e-03 2.73e-02 -1.03e-01
Time step 3540
5903: -3.99e+00 -4.62e+00 -6.77e+00
16745: 2.99e-03 2.07e-02 -1.10e-01
Time step 3570
5903: -3.99e+00 -4.62e+00 -6.77e+00
16745: 4.11e-03 1.27e-02 -1.14e-01
Time step 3600
5903: -3.99e+00 -4.62e+00 -6.77e+00
16745: 4.60e-03 5.60e-03 -1.17e-01
Time step 3630
5903: -3.98e+00 -4.62e+00 -6.77e+00

```

16745: 4.81e-03 1.51e-03 -1.17e-01
Time step 3660
5903: -3.98e+00 -4.62e+00 -6.77e+00
16745: 5.30e-03 1.42e-03 -1.14e-01
Time step 3690
5903: -3.98e+00 -4.62e+00 -6.77e+00
16745: 6.40e-03 4.99e-03 -1.11e-01
Time step 3720
5903: -3.98e+00 -4.62e+00 -6.76e+00
16745: 7.90e-03 1.08e-02 -1.06e-01
Time step 3750
5903: -3.98e+00 -4.62e+00 -6.76e+00
16745: 9.03e-03 1.70e-02 -1.03e-01
Time step 3780
5903: -3.98e+00 -4.62e+00 -6.76e+00
16745: 8.83e-03 2.21e-02 -1.01e-01
Time step 3810
5903: -3.98e+00 -4.62e+00 -6.76e+00
16745: 6.61e-03 2.52e-02 -1.00e-01
Time step 3840
5903: -3.98e+00 -4.62e+00 -6.76e+00
16745: 2.45e-03 2.66e-02 -1.01e-01
equake00: 30169 nodes 151173 elems 3855 timesteps

```

equake00: Done. Terminating the simulation.

```

sim: ** simulation statistics **
sim_num_insn      165643487723 # total number of instructions executed
sim_num_refs      78603143990 # total number of loads and stores executed
sim_elapsed_time   7607 # total simulation time in seconds
sim_inst_rate     21775139.7033 # simulation speed (in insts/sec)
ld_text_base      0x00400000 # program text (code) segment base
ld_text_size      132784 # program text (code) size in bytes
ld_data_base      0x10000000 # program initialized data segment base
ld_data_size      16384 # program init'ed '.data' and uninit'ed '.bss' size in bytes
ld_stack_base     0x7fff0000 # program stack segment base (highest address in stack)
ld_stack_size     16384 # program initial stack size
ld_prog_entry     0x00400140 # program entry point (initial PC)
ld_envirion_base  0x7fff8000 # program environment base address address
ld_target_big_endian 0 # target executable endianness, non-zero if big endian
mem.page_count    10395 # total number of pages allocated
mem.page_mem      41580k # total size of memory pages allocated
mem.ptab_misses   3253142 # total first level page table misses
mem.ptab_accesses 906494438806 # total page table accesses
mem.ptab_miss_rate 0.0000 # first level page table miss rate

```

Thus, we have total 165643487723 instructions in this program.

a. Estimation about real running time at a 3GHz, 1.6747 IPC processor

$$Running\ time = \frac{IC}{CPU\ Frequency * IPC} = \frac{165643487723(instruction)}{3 \times 10^9(\frac{cycles}{sec}) * 1.6747(\frac{instruction}{cycles})} \approx 32.97(sec)$$

b. Estimation about simulation running time at my own PC

From question 1, I know that the simulation speed at my own computer is $1395348.8419(\frac{instruction}{sec})$.

Although this number is only meaningful to the 300 million instructions run in the question 1, I assume this

number is a good estimation for the whole program. Thus, $Simulation\ time = \frac{IC}{Simulation\ Speed} =$

$$\frac{165643487723(instruction)}{1395348.8419(\frac{instruction}{sec})} \approx 118711.1658(sec) \approx 32.98\ (hour)$$

4. Memory and cache access time analysis

a. Default configuration

Please see the detail results at section 1 (P1-P3). Latency of level 1 cache is 1 cycle. Latency of level 2 cache is 6 cycles. Latency of memory is 18 cycles for the first chunk and 2 cycles for the rest.

b. Infinitely large L2 cache

i. Command

As the requirement says, once the memory access latency is equal to L2 cache hit latency, L2 cache can be considered as infinitely large. Unfortunately, this is just a hypothesis that can only be true if the simpliscalar source files use these two latency exactly the same way and set both the access functionality of memory module and cache module exactly the same. Thus, it's necessary to take a look at the way simpliscalar deal with these two modules. The source codes are contained at "sim-outorder.c", "cache.c" and "memory.c".

First, in the "sim-outorder.c", as showed below, when the required data is in the dl2, the program should call "cache_access()" to return the true cache hit latency as stated in line 440. When the required is in the memory, the program should call "mem_access_latency()" to return the main memory access latency as stated in line 454.

```

426
427 /* l1 data cache l1 block miss handler function */
428 static unsigned int /* latency of block access */
429 dl1_access_fn(enum mem cmd cmd, /* access cmd, Read or Write */
430 md addr t baddr, /* block address to access */
431 int bsize, /* size of block to access */
432 struct cache_blk t *blk, /* ptr to block in upper level */
433 tick_t now) /* time of access */
434 {
435     unsigned int lat;
436
437     if (cache_dl2)
438     {
439         /* access next level of data cache hierarchy */
440         lat = cache_access(cache_dl2, cmd, baddr, NULL, bsize,
441 /* now */now, /* padata */NULL, /* repl addr */NULL);
442         if (cmd == Read)
443             return lat;
444         else
445         {
446             /* FIXME: unlimited write buffers */
447             return 0;
448         }
449     }
450     else
451     {
452         /* access main memory */
453         if (cmd == Read)
454             return mem_access_latency(bsize);
455         else
456         {
457             /* FIXME: unlimited write buffers */
458             return 0;
459         }
460     }
461 }

```

Pic. 1 Codes of access latency in sim-outorder.c

So, the next thing to check is that if the "cache_access()" function and the "mem_access_latency()" function work in the same way.

```

410 /* memory access latency, assumed to not cross a page boundary */
411 static unsigned int /* total latency of access */
412 mem_access_latency(int blk_sz) /* block size accessed */
413 {
414     int chunks = (blk_sz + (mem_bus_width - 1)) / mem_bus_width;
415     assert(chunks > 0);
416     return (/* first chunk latency */mem_lat[0] +
417 /* remainder chunk latency */mem_lat[1] * (chunks - 1));
418 }

```

Pic. 2 Codes of mem_access_latency()

```

655
656 cache_hit: /* slow hit handler */
657
658 /* **HIT** */
659 cp->hits++;
660
661 /* copy data out of cache block, if block exists */
662 if (cp->ballocc)
663 {
664     CACHE_BCOPY(cmd, blk, bofs, p, nbytes);
665 }
666
667 /* update dirty status */
668 if (cmd == Write)
669     blk->status |= CACHE_BLK_DIRTY;
670
671 /* if LRU replacement and this is not the first element of list, reorder */
672 if (blk->way_prev && cp->policy == LRU)
673 {
674     /* move this block to head of the way (MRU) list */
675     update_way_list(&cp->sets[set], blk, Head);
676 }
677
678 /* tag is unchanged, so hash links (if they exist) are still valid */
679
680 /* record the last block to hit */
681 cp->last_tagset = CACHE_TAGSET(cp, addr);
682 cp->last_blk = blk;
683
684 /* get user block data, if requested and it exists */
685 if (udata)
686     *udata = blk->user_data;
687
688 /* return first cycle data is available to access */
689 return (int) MAX(cp->hit_latency, (blk->ready - now));

```

Apparently, the access latency in memory depends on data block size, while the latency in cache is either the given hit latency or the time when first cycle data is available to access. Thus, the change of memory first chunk latency and remainder chunk latency is not enough to match the latency of L2 cache. But, as least we can consider it as an estimation.

In order to make this estimation more accurate, I decided to set the first chunk latency the same as L2 cache hit latency and the remainder chunk latency to the smallest value the program allows, which is 1.

Thus, the final command I choose is: `sim-outorder -fastfwd 200000000 -max:inst 300000000 -mem:lat 6 1 earthquake.ss<equake.in`

Pic. 3 Partial codes of `cache_access()`

ii. System setup

sim: simulation started @ Sun Mar 27 20:26:28 2016, options follow:

```
# -config                # load configuration from a file
# -dumpconfig           # dump configuration to a file
# -h                    false # print help message
# -v                    false # verbose operation
# -d                    false # enable debug message
# -i                    false # start in Dlite debugger
-seed                   1 # random number generator seed (0 for timer seed)
# -q                    false # initialize and terminate immediately
# -chkpt                <null> # restore EIO trace execution from <fname>
# -redir:sim            <null> # redirect simulator output to file (non-interactive only)
# -redir:prog          <null> # redirect simulated program output to file
-nice                   0 # simulator scheduling priority
-max:inst              300000000 # maximum number of inst's to execute
-fastfwd              200000000 # number of insts skipped before timing starts
# -ptrace              <null> # generate pipetrace, i.e., <fname>|stdout|stderr> <range>
-fetch:ifqsize         4 # instruction fetch queue size (in insts)
-fetch:mplat           3 # extra branch mis-prediction latency
-fetch:speed           1 # speed of front-end of machine relative to execution core
-bpred                bimod # branch predictor type {nottaken|taken|perfect|bimod|2lev|comb}
-bpred:bimod          2048 # bimodal predictor config (<table_size>)
-bpred:2lev           1 1024 8 0 # 2-level predictor config (<l1size> <l2size> <hist_size> <xor>)
-bpred:comb           1024 # combining predictor config (<meta_table_size>)
-bpred:ras             8 # return address stack size (0 for no return stack)
-bpred:btb            512 4 # BTB config (<num_sets> <associativity>)
# -bpred:spec_update   <null> # speculative predictors update in {ID|WB} (default non-spec)
-decode:width          4 # instruction decode B/W (insts/cycle)
-issue:width           4 # instruction issue B/W (insts/cycle)
-issue:inorder         false # run pipeline with in-order issue
-issue:wrongpath       true # issue instructions down wrong execution paths
-commit:width          4 # instruction commit B/W (insts/cycle)
-ruu:size              16 # register update unit (RUU) size
-lsq:size              8 # load/store queue (LSQ) size
-cache:dl1             dl1:128:32:4:l # l1 data cache config, i.e., {<config>|none}
-cache:dl1lat          1 # l1 data cache hit latency (in cycles)
-cache:dl2             ul2:1024:64:4:l # l2 data cache config, i.e., {<config>|none}
-cache:dl2lat          6 # l2 data cache hit latency (in cycles)
-cache:il1             il1:512:32:1:l # l1 inst cache config, i.e., {<config>|dl1|dl2|none}
-cache:il1lat          1 # l1 instruction cache hit latency (in cycles)
-cache:il2             dl2 # l2 instruction cache config, i.e., {<config>|dl2|none}
-cache:il2lat          6 # l2 instruction cache hit latency (in cycles)
-cache:flush           false # flush caches on system calls
-cache:icompress       false # convert 64-bit inst addresses to 32-bit inst equivalents
-mem:lat               6 1 # memory access latency (<first_chunk> <inter_chunk>)
-mem:width             8 # memory access bus width (in bytes)
-tlb:itlb              itlb:16:4096:4:l # instruction TLB config, i.e., {<config>|none}
-tlb:dtlb              dtlb:32:4096:4:l # data TLB config, i.e., {<config>|none}
-tlb:lat               30 # inst/data TLB miss latency (in cycles)
-res:ialu              4 # total number of integer ALU's available
-res:imult             1 # total number of integer multiplier/dividers available
-res:memport           2 # total number of memory system ports available (to CPU)
-res:fpalu             4 # total number of floating point ALU's available
-res:fpmult            1 # total number of floating point multiplier/dividers available
# -pcstat              <null> # profile stat(s) against text addr's (mult uses ok)
-bugcompat             false # operate in backward-compatible bugs mode (for testing only)
```

Now the latency of memory is roughly equal to L2 cache.

iii. Program output

sim: ** fast forwarding 200000000 insts **

equake00: Reading nodes.
equake00: Reading elements.
sim: ** starting performance simulation **

iv. Performance

```

sim: ** simulation statistics **
sim_num_insn          300000001 # total number of instructions committed
sim_num_refs          98046368 # total number of loads and stores committed
sim_num_loads         68339475 # total number of loads committed
sim_num_stores        29706893.0000 # total number of stores committed
sim_num_branches      78914237 # total number of branches committed
sim_elapsed_time      212 # total simulation time in seconds
sim_inst_rate         1415094.3443 # simulation speed (in insts/sec)
sim_total_insn        316921193 # total number of instructions executed
sim_total_refs        103629218 # total number of loads and stores executed
sim_total_loads       72570922 # total number of loads executed
sim_total_stores      31058296.0000 # total number of stores executed
sim_total_branches    82604307 # total number of branches executed
sim_cycle             179023915 # total simulation time in cycles
sim_IPC               1.6758 # instructions per cycle
sim_CPI               0.5967 # cycles per instruction
sim_exec_BW           1.7703 # total instructions (mis-spec + committed) per cycle
sim_IPB               3.8016 # instruction per branch
IFQ_count             453978907 # cumulative IFQ occupancy
IFQ_fcount            95536254 # cumulative IFQ full count
ifq_occupancy         2.5359 # avg IFQ occupancy (insn's)
ifq_rate              1.7703 # avg IFQ dispatch rate (insn/cycle)
ifq_latency           1.4325 # avg IFQ occupant latency (cycle's)
ifq_full              0.5337 # fraction of time (cycle's) IFQ was full
RUU_count             1817293727 # cumulative RUU occupancy
RUU_fcount            48796429 # cumulative RUU full count
ruu_occupancy         10.1511 # avg RUU occupancy (insn's)
ruu_rate              1.7703 # avg RUU dispatch rate (insn/cycle)
ruu_latency           5.7342 # avg RUU occupant latency (cycle's)
ruu_full              0.2726 # fraction of time (cycle's) RUU was full
LSQ_count             587584943 # cumulative LSQ occupancy
LSQ_fcount            18324998 # cumulative LSQ full count
lsq_occupancy         3.2822 # avg LSQ occupancy (insn's)
lsq_rate              1.7703 # avg LSQ dispatch rate (insn/cycle)
lsq_latency           1.8540 # avg LSQ occupant latency (cycle's)
lsq_full              0.1024 # fraction of time (cycle's) LSQ was full
sim_slip              2712845138 # total number of slip cycles
avg_sim_slip          9.0428 # the average slip between issue and retirement
bpred_bimod.lookups   83954321 # total number of bpred lookups
bpred_bimod.updates   78914235 # total number of updates
bpred_bimod.addr_hits 77473446 # total number of address-predicted hits
bpred_bimod.dir_hits  77473516 # total number of direction-predicted hits (includes addr-hits)
bpred_bimod.misses    1440719 # total number of misses
bpred_bimod.jr_hits   3599818 # total number of address-predicted hits for JR's
bpred_bimod.jr_seen   3599824 # total number of JR's seen
bpred_bimod.jr_non_ras_hits.PP 449569 # total number of address-predicted hits for non-RAS JR's
bpred_bimod.jr_non_ras_seen.PP 449570 # total number of non-RAS JR's seen
bpred_bimod.bpred_addr_rate 0.9817 # branch address-prediction rate (i.e., addr-hits/updates)
bpred_bimod.bpred_dir_rate  0.9817 # branch direction-prediction rate (i.e., all-hits/updates)
bpred_bimod.bpred_jr_rate   1.0000 # JR address-prediction rate (i.e., JR addr-hits/JRs seen)
bpred_bimod.bpred_jr_non_ras_rate.PP 1.0000 # non-RAS JR addr-pred rate (ie, non-RAS JR hits/JRs seen)
bpred_bimod.retstack_pushes 3241508 # total number of address pushed onto ret-addr stack
bpred_bimod.retstack_pops   3150293 # total number of address popped off of ret-addr stack
bpred_bimod.used_ras.PP     3150254 # total number of RAS predictions used
bpred_bimod.ras_hits.PP     3150249 # total number of RAS hits
bpred_bimod.ras_rate.PP     1.0000 # RAS prediction rate (i.e., RAS hits/used RAS)
il1.accesses           329973086 # total number of accesses
il1.hits               322322364 # total number of hits
il1.misses             7650722 # total number of misses
il1.replacements       7650507 # total number of replacements
il1.writebacks         0 # total number of writebacks
il1.invalidations      0 # total number of invalidations
il1.miss_rate          0.0232 # miss rate (i.e., misses/ref)
il1.repl_rate          0.0232 # 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)
dl1.accesses           99036937 # total number of accesses
dl1.hits               98967530 # total number of hits
dl1.misses             69407 # total number of misses
dl1.replacements       68895 # total number of replacements
dl1.writebacks         56171 # total number of writebacks

```

dl1.invalidations	0 # total number of invalidations
dl1.miss_rate	0.0007 # miss rate (i.e., misses/ref)
dl1.repl_rate	0.0007 # replacement rate (i.e., repls/ref)
dl1.wb_rate	0.0006 # writeback rate (i.e., wrbks/ref)
dl1.inv_rate	0.0000 # invalidation rate (i.e., invs/ref)
ul2.accesses	7776300 # total number of accesses
ul2.hits	7742316 # total number of hits
ul2.misses	33984 # total number of misses
ul2.replacements	29888 # total number of replacements
ul2.writebacks	24574 # total number of writebacks
ul2.invalidations	0 # total number of invalidations
ul2.miss_rate	0.0044 # miss rate (i.e., misses/ref)
ul2.repl_rate	0.0038 # replacement rate (i.e., repls/ref)
ul2.wb_rate	0.0032 # writeback rate (i.e., wrbks/ref)
ul2.inv_rate	0.0000 # invalidation rate (i.e., invs/ref)
itlb.accesses	329973086 # total number of accesses
itlb.hits	329973074 # total number of hits
itlb.misses	12 # total number of misses
itlb.replacements	0 # total number of replacements
itlb.writebacks	0 # total number of writebacks
itlb.invalidations	0 # total number of invalidations
itlb.miss_rate	0.0000 # miss rate (i.e., misses/ref)
itlb.repl_rate	0.0000 # replacement rate (i.e., repls/ref)
itlb.wb_rate	0.0000 # writeback rate (i.e., wrbks/ref)
itlb.inv_rate	0.0000 # invalidation rate (i.e., invs/ref)
dtlb.accesses	100835273 # total number of accesses
dtlb.hits	100834736 # total number of hits
dtlb.misses	537 # total number of misses
dtlb.replacements	409 # total number of replacements
dtlb.writebacks	0 # total number of writebacks
dtlb.invalidations	0 # total number of invalidations
dtlb.miss_rate	0.0000 # miss rate (i.e., misses/ref)
dtlb.repl_rate	0.0000 # replacement rate (i.e., repls/ref)
dtlb.wb_rate	0.0000 # writeback rate (i.e., wrbks/ref)
dtlb.inv_rate	0.0000 # invalidation rate (i.e., invs/ref)
sim_invalid_addr	0 # total non-speculative bogus addresses seen (debug var)
ld_text_base	0x00400000 # program text (code) segment base
ld_text_size	132784 # program text (code) size in bytes
ld_data_base	0x10000000 # program initialized data segment base
ld_data_size	16384 # program init'ed '.data' and uninit'ed '.bss' size in bytes
ld_stack_base	0x7fffc000 # program stack segment base (highest address in stack)
ld_stack_size	16384 # program initial stack size
ld_prog_entry	0x00400140 # program entry point (initial PC)
ld_environ_base	0x7fff8000 # program environment base address address
ld_target_big_endian	0 # target executable endian-ness, non-zero if big endian
mem.page_count	1176 # total number of pages allocated
mem.page_mem	4704k # total size of memory pages allocated
mem.ptab_misses	6280 # total first level page table misses
mem.ptab_accesses	3321066762 # total page table accesses
mem.ptab_miss_rate	0.0000 # first level page table miss rate

c. Infinitely large L1 cache

i. Command

```
sim-outorder -fastfwd 200000000 -max:inst 300000000 -mem:lat 1 1 -cache:il2lat 1 -cache:dl2lat 1 equake.ss<equake.in
```

ii. System setup

sim: simulation started @ Sun Mar 27 15:37:36 2016, options follow:

# -config	# load configuration from a file
# -dumpconfig	# dump configuration to a file
# -h	false # print help message
# -v	false # verbose operation
# -d	false # enable debug message
# -i	false # start in Dlite debugger
-seed	1 # random number generator seed (0 for timer seed)
# -q	false # initialize and terminate immediately
# -chkpt	<null> # restore EIO trace execution from <fname>
# -redir:sim	<null> # redirect simulator output to file (non-interactive only)
# -redir:prog	<null> # redirect simulated program output to file
-nice	0 # simulator scheduling priority
-max:inst	300000000 # maximum number of inst's to execute


```

-fastfwd          200000000 # number of insts skipped before timing starts
# -ptrace          <null> # generate pipetrace, i.e., <fname|stdout|stderr> <range>
-fetch:ifqsize    4 # instruction fetch queue size (in insts)
-fetch:mplat      3 # extra branch mis-prediction latency
-fetch:speed      1 # speed of front-end of machine relative to execution core
-bpred            bimod # branch predictor type {nottaken|taken|perfect|bimod|2lev|comb}
-bpred:bimod      2048 # bimodal predictor config (<table_size>)
-bpred:2lev       1 1024 8 0 # 2-level predictor config (<l1size> <l2size> <hist_size> <xor>)
-bpred:comb       1024 # combining predictor config (<meta_table_size>)
-bpred:ras        8 # return address stack size (0 for no return stack)
-bpred:btb        512 4 # BTB config (<num_sets> <associativity>)
# -bpred:spec_update <null> # speculative predictors update in {ID|WB} (default non-spec)
-decode:width     4 # instruction decode B/W (insts/cycle)
-issue:width      4 # instruction issue B/W (insts/cycle)
-issue:inorder    false # run pipeline with in-order issue
-issue:wrongpath  true # issue instructions down wrong execution paths
-commit:width     4 # instruction commit B/W (insts/cycle)
-ruu:size         16 # register update unit (RUU) size
-lsq:size         8 # load/store queue (LSQ) size
-cache:dl1        dl1:128:32:4:l # l1 data cache config, i.e., {<config>|none}
-cache:dl1lat     1 # l1 data cache hit latency (in cycles)
-cache:dl2        ul2:1024:64:4:l # l2 data cache config, i.e., {<config>|none}
-cache:dl2lat     1 # l2 data cache hit latency (in cycles)
-cache:il1        il1:512:32:1:l # l1 inst cache config, i.e., {<config>|dl1|dl2|none}
-cache:il1lat     1 # l1 instruction cache hit latency (in cycles)
-cache:il2        dl2 # l2 instruction cache config, i.e., {<config>|dl2|none}
-cache:il2lat     1 # l2 instruction cache hit latency (in cycles)
-cache:flush      false # flush caches on system calls
-cache:icompress  false # convert 64-bit inst addresses to 32-bit inst equivalents
-mem:lat          1 1 # memory access latency (<first_chunk> <inter_chunk>)
-mem:width        8 # memory access bus width (in bytes)
-tlb:itlb         itlb:16:4096:4:l # instruction TLB config, i.e., {<config>|none}
-tlb:dtlb         dtlb:32:4096:4:l # data TLB config, i.e., {<config>|none}
-tlb:lat          30 # inst/data TLB miss latency (in cycles)
-res:ialu         4 # total number of integer ALU's available
-res:imult        1 # total number of integer multiplier/dividers available
-res:mempport     2 # total number of memory system ports available (to CPU)
-res:fpalu        4 # total number of floating point ALU's available
-res:fpmult       1 # total number of floating point multiplier/dividers available
# -pcstat          <null> # profile stat(s) against text addr's (mult uses ok)
-bugcompat        false # operate in backward-compatible bugs mode (for testing only)

```

iii. Program output

```

sim: ** fast forwarding 200000000 insts **
quake00: Reading nodes.
quake00: Reading elements.
sim: ** starting performance simulation **

```

iv. Performance

```

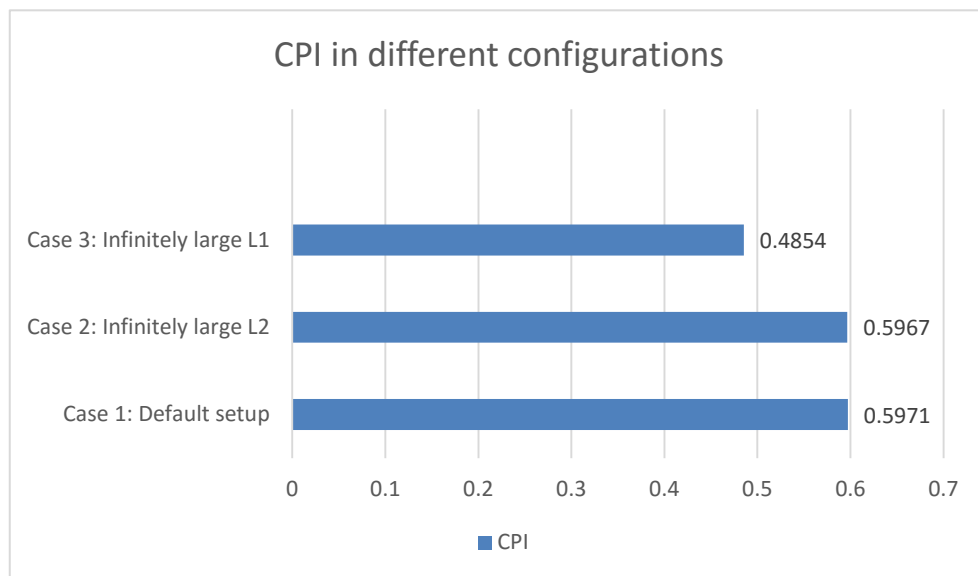
sim: ** simulation statistics **
sim_num_insn      300000001 # total number of instructions committed
sim_num_refs      98046368 # total number of loads and stores committed
sim_num_loads     68339475 # total number of loads committed
sim_num_stores    29706893.0000 # total number of stores committed
sim_num_branches  78914237 # total number of branches committed
sim_elapsed_time  232 # total simulation time in seconds
sim_inst_rate     1293103.4526 # simulation speed (in insts/sec)
sim_total_insn    317461070 # total number of instructions executed
sim_total_refs    103988881 # total number of loads and stores executed
sim_total_loads   72570926 # total number of loads executed
sim_total_stores  31417955.0000 # total number of stores executed
sim_total_branches 82604392 # total number of branches executed
sim_cycle         145612215 # total simulation time in cycles
sim_IPC           2.0603 # instructions per cycle
sim_CPI           0.4854 # cycles per instruction
sim_exec_BW       2.1802 # total instructions (mis-spec + committed) per cycle
sim_IPB           3.8016 # instruction per branch
IFQ_count         496922965 # cumulative IFQ occupancy
IFQ_fcunt         106768813 # cumulative IFQ full count
ifq_occupancy     3.4126 # avg IFQ occupancy (insn's)
ifq_rate          2.1802 # avg IFQ dispatch rate (insn/cycle)
ifq_latency       1.5653 # avg IFQ occupant latency (cycle's)
ifq_full          0.7332 # fraction of time (cycle's) IFQ was full
RUU_count         1894766307 # cumulative RUU occupancy

```

RUU_fcount	64937243 # cumulative RUU full count
ruu_occupancy	13.0124 # avg RUU occupancy (insn's)
ruu_rate	2.1802 # avg RUU dispatch rate (insn/cycle)
ruu_latency	5.9685 # avg RUU occupant latency (cycle's)
ruu_full	0.4460 # fraction of time (cycle's) RUU was full
LSQ_count	615506792 # cumulative LSQ occupancy
LSQ_fcount	18330095 # cumulative LSQ full count
lsq_occupancy	4.2270 # avg LSQ occupancy (insn's)
lsq_rate	2.1802 # avg LSQ dispatch rate (insn/cycle)
lsq_latency	1.9388 # avg LSQ occupant latency (cycle's)
lsq_full	0.1259 # fraction of time (cycle's) LSQ was full
sim_slip	2816801952 # total number of slip cycles
avg_sim_slip	9.3893 # the average slip between issue and retirement
bpred_bimod.lookups	84044465 # total number of bpred lookups
bpred_bimod.updates	78914235 # total number of updates
bpred_bimod.addr_hits	77473446 # total number of address-predicted hits
bpred_bimod.dir_hits	77473518 # total number of direction-predicted hits (includes addr-hits)
bpred_bimod.misses	1440717 # total number of misses
bpred_bimod.jr_hits	3599818 # total number of address-predicted hits for JR's
bpred_bimod.jr_seen	3599824 # total number of JR's seen
bpred_bimod.jr_non_ras_hits.PP	449569 # total number of address-predicted hits for non-RAS JR's
bpred_bimod.jr_non_ras_seen.PP	449570 # total number of non-RAS JR's seen
bpred_bimod.bpred_addr_rate	0.9817 # branch address-prediction rate (i.e., addr-hits/updates)
bpred_bimod.bpred_dir_rate	0.9817 # branch direction-prediction rate (i.e., all-hits/updates)
bpred_bimod.bpred_jr_rate	1.0000 # JR address-prediction rate (i.e., JR addr-hits/JRs seen)
bpred_bimod.bpred_jr_non_ras_rate.PP	1.0000 # non-RAS JR addr-pred rate (ie, non-RAS JR hits/JRs seen)
bpred_bimod.retstack_pushes	3331424 # total number of address pushed onto ret-addr stack
bpred_bimod.retstack_pops	3150293 # total number of address popped off of ret-addr stack
bpred_bimod.used_ras.PP	3150254 # total number of RAS predictions used
bpred_bimod.ras_hits.PP	3150249 # total number of RAS hits
bpred_bimod.ras_rate.PP	1.0000 # RAS prediction rate (i.e., RAS hits/used RAS)
il1.accesses	324121876 # total number of accesses
il1.hits	316471153 # total number of hits
il1.misses	7650723 # total number of misses
il1.replacements	7650508 # total number of replacements
il1.writebacks	0 # total number of writebacks
il1.invalidations	0 # total number of invalidations
il1.miss_rate	0.0236 # miss rate (i.e., misses/ref)
il1.repl_rate	0.0236 # 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)
dl1.accesses	99036875 # total number of accesses
dl1.hits	98967466 # total number of hits
dl1.misses	69409 # total number of misses
dl1.replacements	68897 # total number of replacements
dl1.writebacks	56171 # total number of writebacks
dl1.invalidations	0 # total number of invalidations
dl1.miss_rate	0.0007 # miss rate (i.e., misses/ref)
dl1.repl_rate	0.0007 # replacement rate (i.e., repls/ref)
dl1.wb_rate	0.0006 # writeback rate (i.e., wrbks/ref)
dl1.inv_rate	0.0000 # invalidation rate (i.e., invs/ref)
ul2.accesses	7776303 # total number of accesses
ul2.hits	7742317 # total number of hits
ul2.misses	33986 # total number of misses
ul2.replacements	29890 # total number of replacements
ul2.writebacks	24574 # total number of writebacks
ul2.invalidations	0 # total number of invalidations
ul2.miss_rate	0.0044 # miss rate (i.e., misses/ref)
ul2.repl_rate	0.0038 # replacement rate (i.e., repls/ref)
ul2.wb_rate	0.0032 # writeback rate (i.e., wrbks/ref)
ul2.inv_rate	0.0000 # invalidation rate (i.e., invs/ref)
itlb.accesses	324121876 # total number of accesses
itlb.hits	324121863 # total number of hits
itlb.misses	13 # total number of misses
itlb.replacements	0 # total number of replacements
itlb.writebacks	0 # total number of writebacks
itlb.invalidations	0 # total number of invalidations
itlb.miss_rate	0.0000 # miss rate (i.e., misses/ref)
itlb.repl_rate	0.0000 # replacement rate (i.e., repls/ref)
itlb.wb_rate	0.0000 # writeback rate (i.e., wrbks/ref)
itlb.inv_rate	0.0000 # invalidation rate (i.e., invs/ref)
dtlb.accesses	100835154 # total number of accesses
dtlb.hits	100834617 # total number of hits
dtlb.misses	537 # total number of misses
dtlb.replacements	409 # total number of replacements
dtlb.writebacks	0 # total number of writebacks
dtlb.invalidations	0 # total number of invalidations
dtlb.miss_rate	0.0000 # miss rate (i.e., misses/ref)

dtlb.repl_rate	0.0000 # replacement rate (i.e., repls/ref)
dtlb.wb_rate	0.0000 # writeback rate (i.e., wrbks/ref)
dtlb.inv_rate	0.0000 # invalidation rate (i.e., invs/ref)
sim_invalid_addrs	0 # total non-speculative bogus addresses seen (debug var)
ld_text_base	0x00400000 # program text (code) segment base
ld_text_size	132784 # program text (code) size in bytes
ld_data_base	0x10000000 # program initialized data segment base
ld_data_size	16384 # program init'ed '.data' and uninit'ed '.bss' size in bytes
ld_stack_base	0x7fffc000 # program stack segment base (highest address in stack)
ld_stack_size	16384 # program initial stack size
ld_prog_entry	0x00400140 # program entry point (initial PC)
ld_enviro_base	0x7fff8000 # program environment base address address
ld_target_big_endian	0 # target executable endian-ness, non-zero if big endian
mem.page_count	1176 # total number of pages allocated
mem.page_mem	4704k # total size of memory pages allocated
mem.ptab_misses	6280 # total first level page table misses
mem.ptab_accesses	3297661928 # total page table accesses
mem.ptab_miss_rate	0.0000 # first level page table miss rate

Comments:



From the chart above, we can clearly see that CPI increasing from case 3 to case 1.

First, let's compare case 3 with case 2. In this condition, infinitely large L2 cache replaces infinitely large L1 cache and this replacement increases CPI value from 0.4854 to 0.5967. This indicates the performance of L2 cache is worse than L1 cache.

Second, consider case 2 with case 1. The same thing happens again. CPI grows from 0.5967 to 0.5971 so that memory is the weakest part of whole execution time.

All in all, memory is the bottleneck for this program running on the simulated machine.