Introduction to SimpleScalar

CMPEN 431

Overview

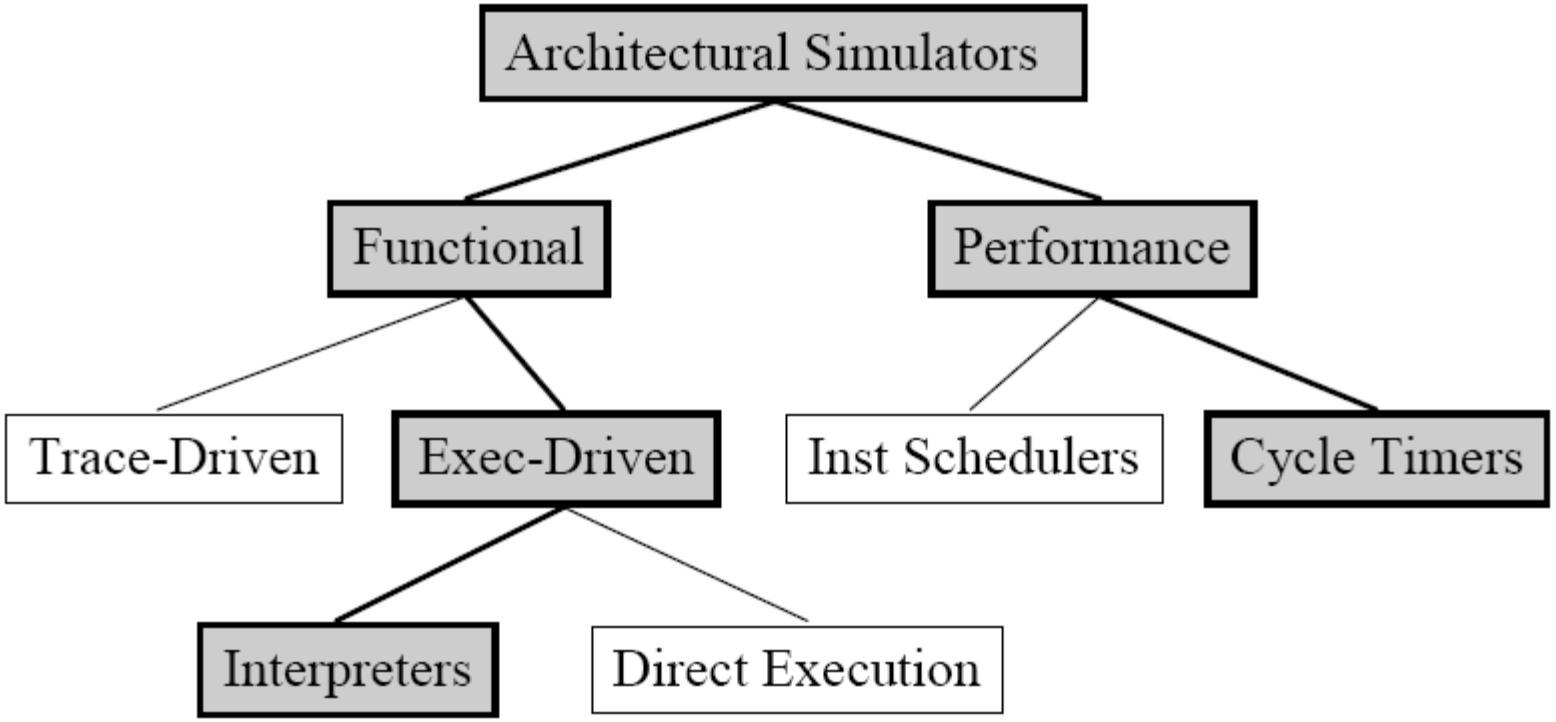
- Computer Architecture Simulator
- SimpleScalar
- Environment Setup
- Remote Access

Simulator

- What is an architectural simulator?
  - is a piece of software to model computer devices (or components) to predict outputs and performance metrics on a given input(From Wikipedia)

- Why we use a simulator?
  - Leverage a faster, more flexible software development cycle
    - Permit more design space exploration
    - Facilitate validation before HW becomes available
    - Level of abstraction is tailored by design task
    - Possible to increase/improve system instrumentation
    - Usually less expensive than building a real system

A Taxonomy of Simulation Tools



Shaded tools are included in SimpleScalar Tool Set

Functional vs. Performance

- **Functional simulators**
  - Model the function of each component
  - Does not have timing accuracy
  - Faster
- **Performance simulators**
  - Model system at finer detail
  - Timing in addition to functionality
- For example:
  - Branch prediction accuracy vs. memory latency

Overview

- Computer Architecture Simulator
- SimpleScalar
- Environment Setup
- Remote Access

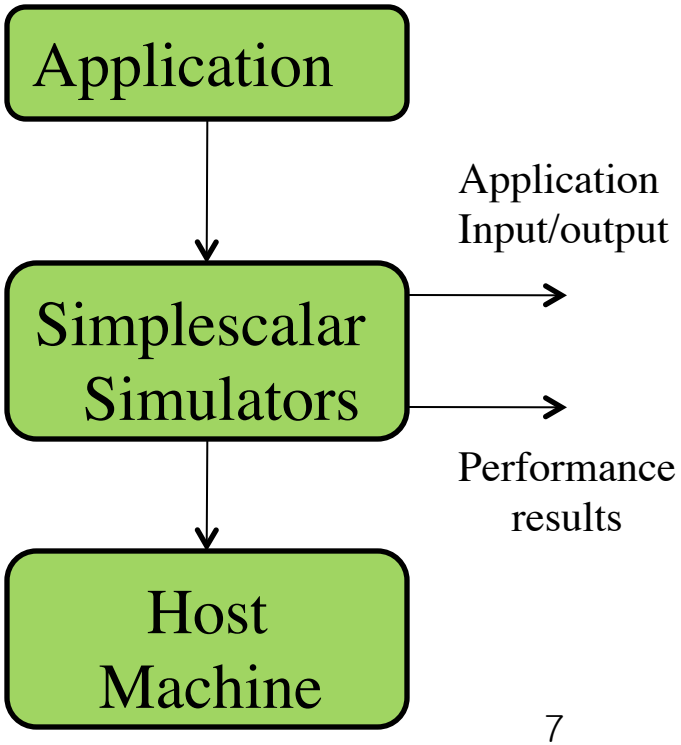
SimpleScalar Tool Set

- Computer system design and analysis infrastructure
  - Processor/device (behavioral) models

    build modeling applications that simulate real programs running on a range of modern processors and systems
  - Support many ISAs and I/O interfaces

    Alpha, PISA, ARM, and x86 instr. sets
  - Hosted on most any Unix-like machine

- Freely available for academic non-commercial use with source from www.simplescalar.com



Advantages of SimpleScalar

- Highly flexible
  - functional simulator + performance simulator
- Portable
  - Host: virtual target runs on most Unix-like systems
  - Target: simulators can support multiple ISAs
- Extensible
  - Source is included for compiler, libraries, simulators
  - Easy to write simulators
- Performance
  - Sim-Fast: 10+ MIPS
  - Sim-OutOrder: 350+ KIPS

Simulator Suite

Sim-Fast	Sim-Safe	Sim-Profile	Sim-Cache Sim-BPred	Sim-Outorder
-Less than 300 lines -functional -4+ MIPS	-Less than 300 lines -functional w/checks	-900 lines -functional -Lot of stats	-~1000 lines -functional -Cache stats	-3900 lines -performance -OoO issue -Branch pred. -Mis-spec. -ALUs -Cache -TLB

Performance ←

Detail →

Sim-Fast

- Functional simulation
- Optimized for speed
- Assumes no cache
- Assumes no instruction checking
- Does not allow command line arguments
- <300 lines of code

Sim-Cache

- Cache simulation
- Ideal for fast simulation of caches (if the effect of cache performance on execution time is not necessary)
- Accepts command line arguments for:
  - level 1 & 2 instruction and data caches
  - TLB configuration (data and instruction)
  - Flush and compress
  - and more
- Ideal for performing high-level cache studies that don't take access time of the caches into account

Sim-Bpred

- Simulate different branch prediction mechanisms
- Generate prediction hit and miss rate reports
- Does not simulate the effect of branch prediction on total execution time

nottaken	
taken	
perfect	
bimod	bimodal predictor
2lev	2-level adaptive predictor
comb	combined predictor (bimodal and 2-level)

Sim-Outorder

- Most complicated and detailed simulator
- Supports out-of-order issue and execution
- Provides reports
  - branch prediction
  - cache
  - external memory
  - various configuration

Specifying Sim-outorder

- **Running a program**

  sim-outorder [sim opts] program [program opts]

- **e.g.**

$SIMPLESIM/simplesim-3.0/sim-outorder

-config cfg_file bzip2_base.i386-m32-gcc42-nn
dryer.jpg

Overview

- Computer Architecture Simulator
- SimpleScalar
- Environment Setup
- Remote Access

Benchmark

- SPEC 2006
  - Six benchmarks (4 integer, 2 floating point)

    bzip2(INT)

    equake(FP)

    hmmer(INT)

    mcf(INT)

    milc(FP)

    sjeng(INT)

Installation of simplescalar

- Simplescalar 3.0 is already installed in CSE lab 218
- Path setup

  1) log on into one of the linux machines

  2) Go to your home directory: cd

  3) > vim .cshrc (or gedit .cshrc)

  setenv SIMPLESIM /home/software/simplesim

  > source .cshrc

  4) to verify, run

  > echo $SIMPLESIM

  the return should be /home/software/simplesim

Installation of simplescalar

- create a local directory

  >mkdir simplescalar

  >cd simplescalar

  >cp –r $SIMPLESIM/ss-benchmark .

  >cd ss-benchmark

- Download tmp.cfg from ANGEL
  - Save it in /ss-benchmark

Running Benchmarks

- Run benchmark (bzip2)

  >cd bzip2

  >$SIMPLESIM/simplesim-3.0/sim-outorder –config ../**tmp.cfg** bzip2_base.i386-m32-gcc42-nn dryer.jpg

Check results

- Check simulation results
  - vim sim1.out (or gedit sim1.out)

```
sim: ** fast forwarding 300000 insts **
sim: ** starting performance simulation **

sim: ** simulation statistics **
sim_num_insn                 2000000 # total number of instructions committed
sim_num_refs                  711143 # total number of loads and stores committed
sim_num_loads                 306852 # total number of loads committed
sim_num_stores           404291.0000 # total number of stores committed
sim_num_branches              212047 # total number of branches committed
sim_elapsed_time                   4 # total simulation time in seconds
sim_inst_rate            500000.0000 # simulation speed (in insts/sec)
sim_total_insn               2000000 # total number of instructions executed
sim_total_refs                711143 # total number of loads and stores executed
sim_total_loads               306852 # total number of loads executed
sim_total_stores         404291.0000 # total number of stores executed
sim_total_branches            212047 # total number of branches executed
sim_cycle                    7787523 # total simulation time in cycles
sim_IPC                       0.2568 # instructions per cycle
sim_CPI                       3.8938 # cycles per instruction
sim_exec_BW                   0.2568 # total instructions (mis-spec + committed) per cycle
sim_IPB                       9.4319 # instruction per branch
```

Modify config

- Modify a parameter in the config file

  > cd ..

  > vim tmp.cfg (or gedit tmp.cfg)

  -Increase L2 Data Cache Latency from 4 to 10

  -cache:dl2lat 10

  - Change output file name (-redir:sim sim2.out)

  - Save and close tmp.cfg

Re-run Benchmark

- Run benchmark (bzip2)

```
>cd bzip2
>$SIMPLESIM/simplesim-3.0/sim-outorder
  –config ../tmp.cfg
   bzip2_base.i386-m32-gcc42-nn dryer.jpg
```

Check result

- Check simulation results
  - vim sim2.out (or gedit sim2.out)

```
sim: ** fast forwarding 300000 insts **
sim: ** starting performance simulation **

sim: ** simulation statistics **
sim_num_insn                 2000000 # total number of instructions committed
sim_num_refs                  711143 # total number of loads and stores committed
sim_num_loads                 306852 # total number of loads committed
sim_num_stores           404291.0000 # total number of stores committed
sim_num_branches              212047 # total number of branches committed
sim_elapsed_time                   5 # total simulation time in seconds
sim_inst_rate            400000.0000 # simulation speed (in insts/sec)
sim_total_insn               2000000 # total number of instructions executed
sim_total_refs                711143 # total number of loads and stores executed
sim_total_loads               306852 # total number of loads executed
sim_total_stores         404291.0000 # total number of stores executed
sim_total_branches            212047 # total number of branches executed
sim_cycle                    8683285 # total simulation time in cycles
sim_IPC                       0.2303 # instructions per cycle
sim_CPI                       4.3416 # cycles per instruction
sim_exec_BW                   0.2303 # total instructions (mis-spec + committed) per cycle
sim_IPB                       9.4319 # instruction per branch
```

Global Simulator Options

- **Supported on all simulators**

-h	-print simulator help message
-d	-enable debug message
-q	-quit immediately
-config <file>	-read config parameters from <file>
-dumpconfig <file>	-save config parameters into <file>

- **Configuration files**
  - To generate a configuration file: -dumpconfig <file>
  - Change parameters in <file>
  - comments allowed in configuration file, “#”
  - Reload configuration file using: -config <file>

Get tmp.cfg

- How to generate the input file ?

>$SIMPLESIM/simplesim-3.0/sim-outorder

-dumpconfig tmp1.cfg

vim tmp1.cfg

i

//start editing

Esc

:wq

```
-issue:inorder                false

# issue instructions down wrong execution paths
-issue:wrongpath               true

# instruction commit B/W (insts/cycle)
-commit:width                     4

# register update unit (RUU) size
-ruu:size                        16

# load/store queue (LSQ) size
-lsq:size                         8

# l1 data cache config, i.e., {<config>|none}
-cache:dl1             dl1:128:32:4:l

# l1 data cache hit latency (in cycles)
-cache:dl1lat                     1

# l2 data cache config, i.e., {<config>|none}
-cache:dl2             ul2:1024:64:4:l
                                 103,1         58%
```

Overview

- Computer Architecture Simulator
- SimpleScalar
- Environment Setup
- Remote Access

Log into Lab 218 using SSH

- Visit 2FA.psu.edu and configure a device to receive your second factor codes, or alternatively enable “push” verification.
- Visit https://vpn.cse.psu.edu to install VPN client. This step requires 2 factor authentication.
  - This website requires a CSE login and password, you will need to be on the CSE network
  - Machine names are in the following format: p218instXX.cse.psu.edu, where XX is a number for the machine ID

Running rest of benchmarks

- mcf

  > cd mcf

  > $SIMPLESIM/simplesim-3.0/sim-outorder

  –config ../**tmp.cfg** **mcf_base.i386-m32-gcc42-nn** **inp.in**

- hmmer

  > cd hmmer

  > $SIMPLESIM/simplesim-3.0/sim-outorder

  –config ../**tmp.cfg** **hmmer_base.i386-m32-gcc42-nn** **bombesin.hmm**

- sjeng

  > cd sjeng

  > $SIMPLESIM/simplesim-3.0/sim-outorder

  –config ../**tmp.cfg** **sjeng_base.i386-m32-gcc42-nn** **test.txt**

- milc

  > cd milc

  > $SIMPLESIM/simplesim-3.0/sim-outorder

  –config ../**tmp.cfg milc_base.i386-m32-gcc42-nn** < **su3imp.in**

- equake

  > cd equake

  > $SIMPLESIM/simplesim-3.0/sim-outorder

  –config ../**tmp.cfg** **equake_base.pisa_little** **<** **inp.in** **>** **inp.out**

Files on ANGEL

- This tutorial
- tmp.cfg
- SimpleScalar v2.0 user guide
- Lab environment setup