Tuesday, September 2, 2014

Lecture 2

Topic: Overview of different modeling/evaluation techniques

Desirable Features

- Accurate
- Not expensive
- Non-invasive
  - inserting measurement would introduce overhead
  - desirable: low overhead
- User-friendly
  - generally, more user-friendly tools would be more invasive (e.g., extra codes added to support GUI)
- Fast
- Easy to change or extend
- Must not need application source
  - specific to computer performance evaluation
- Measure all activity
  - specific to computer performance evaluation
  - e.g., not all modeling tools take OS into consideration
- Provide control over aspects measured
  - increase controllability
  - advanced feature, not a necessary

Tradeoffs in modeling/evaluation techniques

- Analytical Models
  - Estimate through calculation. Simplest one: analytical equations

  - Pros: Fast
  - Cons: Not accurate (tends to ignore some details)
- Tools with GUI
  - Pros: User-friendly
  - Cons: Invasive

Simulation

- Focus of this course: Timing simulators
- Timing simulators might or might not implement functions
- Example1 : Cache Simulator
  - For timing simulator: build a model with partial functionality (do not have to move data around) and all design parameters (easy to change and extend)
  - different ways to implement model:
    - implement in C: fast, useful in early design
    - implement in verilog: slow but accurate
  - Accuracy
    - need good assumptions
    - might not be the same as implementation with some details ignored
      - e.g., not consider speculative branches and incorrect prediction overhead
    - ideally, linked to a CPU simulator.
      - Problem: introduce performance variability
- Trace driven simulation
  - Pros:
    - Simple
    - Easy to debug: important because testing is hard, need to go through several runs of debugging to include all corner cases

- Experiments repeatable easily:
  - execution driven simulation: unpredictable because of many causes of variability
- Cons:
  - prohibitively long:
    - reason: proportional to dynamic instruction size (if run a loop for hundreds of times….)
    - solution:
      - trace sampling (focus on big issues; might be biasing)
      - trace reduction/compression (e.g., count how many loops are executed. QPT reference: http://pages.cs.wisc.edu/~larus/qpt.html)
  - mispredicted path missing:
    - solution:
      - in the trace: taken/not taken. —> construct memory image —> get mispredicted path
      - problem with solution: not all mistaken path can be reconstructed

- Example 2: pipelined execution simulation
  - Analytical Model: even simpler than implementing model in C
  - Amadahl's law
    - S = 1/ {f/N + (1-f)} (f: work that could be pipelined; N: depth of pipeline)
    - assumption: no bubbles
    - reality: bubble exists
    - taking bubble into consideration:
      - S = 1 / {(1-g)+(g/N)} (g: equivalent pipelined work; N: # of busy stages = # of stages in pipeline + 1 - cycle penalty)
      - S = 1 / {g1/1 + g2/2 + g3/3 +…gN/N} (consider different causes of bubble separately)
      - Exercise on slides:

        - branches: 66.6% of branches (0.666 * 0.2 = 0.13) would be mispredicted, need to insert bubbles. # of busy stages: 6 - 4 = 2
        - loads: # of busy stages: 6 -1 = 5
        - other instructions: no bubbles needed
- Features of analytical model:
    - fast
    - simple
    - assumptions would affect results (like, the percentage of loads in program)
    - can be applied to various workloads
    - used to narrow down micro-architecture choices
    - used for validation (apply to all simulators because simulators might have errors.)

Classification of Techniques

- Performance Modeling:
    - Simulation:
        - trace-driven simulation (as in example1, simulator does not execute instructions)
        - execution-driven simulation (simulator execute instructions, can solve the problem of speculation information missing)
        - complete system simulation (very accurate, but also very slow)
        - event-driven simulation (look for certain events)
        - statistical simulation
    - Analytical Modeling:
        - most simple one: equations as in example2
    - Performance Measurement:
        - on-chip hardware monitoring:
            - tests at run time

  - use the counters on processors (available and easy to use)
- off-chip hardware monitoring
  - less invasive than on-chip
  - measuring chip sits by the side
- software monitoring
  - measuring software runs parallel at background
- microcoded instructions
  - inserting codes into instructions