Lecture 7 – Estimation and Evaluation

Andreas Gerstlauer
Electrical and Computer Engineering
University of Texas at Austin
gerstl@ece.utexas.edu

Lecture 7: Outline

- Profiling
  - Source-level analysis
  - Retargeting

- Evaluation and estimation
  - Static analysis
  - Simulation
  - Hybrid methods
**Design Space Exploration**

- **Runtime vs. accuracy**
  - Fast design space exploration
  - Fidelity: relative accuracy (vs. absolute accuracy)

- **Capabilities**
  - Various levels of abstraction: components, system
  - Wide range of metrics: power, timing, area, reliability
  - Wide variety of target implementations

---

**Design Space Exploration**

- **Explore and trim**
  - Gradually prune design space

- **Exploration Space**
  - Profiling
  - Impl! independent simulation
  - One-time retargeting
  - Impl! dependent simulation/analysis

- **Time**
  - Profiling stage
  - Retargeting stage
  - Evaluation stage
Evaluation Flow

Refinement | Validation | Feedback
---|---|---
Spec model | Instrumentation | Spec characteristics
Simulation | | Design decision

Profiling

- **Input specification MoC**
  - Hierarchy
  - Computation & communication

- **Multi-dimensional analysis**
  - Multi-entities
    - Behavior, channel, port, variable
  - Multi-metrics
    - Operation, traffic, storage
    - Static, dynamic
  - Multi-levels
    - Application, transaction, bus-functional
Profiling

- **Instrumentation-based profiling**
  - \( B_b \): The execution counts of basic block \( b \)
    - Enumerate execution paths
  - \( C_{b,i,d} \): No. of computed characteristics for item type \( i \)
    and data type \( d \) in the block \( b \)
  - Data type \( i \): float, int, ...
  - Item type \( d \): metric-dependent

  \[ R_{++,int} = \sum_i \left[ B_i \cdot C_{i,++,int} \right] \]
  \[ = 1 \cdot 1 + 3 \cdot 2 \]
  \[ = 7 \]

- **Specification metrics**
  - \( R_{i,d} = \sum_b C_{b,i,d} B_b \)
  - \( R = \sum_d \sum_i R_{i,d} \)


Retargeting

- **Target machine model**
  - \( W_{i,d} \): weights of components which the entity mapped to
    - Manual
    - Simulation
    - Complex cost function/algorithm

- **Implementation estimates**
  - \( E = \sum_i \sum_d (R_{i,d} \cdot W_{i,d}) \)
  - Time complexity: \( O(n) \)

\[ E(B1,PE1)_{++,int}= 7 \times 1 = 7 \]

Vocoder Profiling

Computational complexity of top-level Vocoder behaviors:

<table>
<thead>
<tr>
<th>Behavior</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>LP_Analysis</td>
<td>377.0 MOp</td>
</tr>
<tr>
<td>Open_Loop</td>
<td>337.1 MOp</td>
</tr>
<tr>
<td>Closed_Loop</td>
<td>478.7 MOp</td>
</tr>
<tr>
<td>Codebook</td>
<td>646.4 MOp</td>
</tr>
<tr>
<td>Update</td>
<td>43.6 MOp</td>
</tr>
</tbody>
</table>

Codebook operation mix:

<table>
<thead>
<tr>
<th>Operation</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>(x, int)</td>
<td>46.2%</td>
</tr>
<tr>
<td>(+, int)</td>
<td>33.5%</td>
</tr>
<tr>
<td>(-, int)</td>
<td>9.1%</td>
</tr>
<tr>
<td>(/, int)</td>
<td>7.1%</td>
</tr>
<tr>
<td>(others, int)</td>
<td>4.1%</td>
</tr>
</tbody>
</table>

Floating-point not required
Dedicated hardware multipliers
HW acceleration

Vocoder Design Space Exploration

- Mapping of 8 top-level encoder behaviors onto ColdFire + DSP + HW
- 85.04h for 6561 alternatives (1.7s simulation + 3s refinement each)
- 100% fidelity
Lecture 7: Outline

✓ Profiling
  ✓ Source-level analysis
  ✓ Retargeting

• Evaluation and estimation
  • Static analysis
  • Simulation
  • Hybrid methods

Evaluation and Estimation Methods

• Measurement
  • Fast (real time), exhaustive?
  • Requires physical implementation

• Analysis
  • Worst-case/best-case assumptions
  • Tightness of upper/lower bounds? Dynamic effects?

• Simulation
  • Speed vs. accuracy tradeoffs
  • Quality of testbench, corner cases?

Source: L. Thiele
Estimation Methods

- **Static analysis**
  - Symbolic, mathematical models for best/worst case
    - Worst-case execution time analysis (WCET)
    - Real-time scheduling

- **Probabilistic analysis**
  - Statistical models, distributions for “average” case
    - Queuing theory

- **Deterministic dynamic analysis**
  - Min-plus/max-plus algebra, upper/lower bounds over time
    - Network calculus
    - Real-time calculus
      - Modular Performance Analysis (MPA)

Static Code Analysis

- **Worst-case execution time (WCET)**
  - Micro-architecture analysis
    - Compute bounds for each basic execution block
    - Symbolically simulate statements on processor model (pipeline)
      - Conservative assumptions for dynamic effects (caches, predictors)
  - Path analysis
    - Enumerate possible paths and take maximum of block sequence
      - Possible paths often highly dynamic (loop bounds, false paths)
      - Basis for back-annotation or static system analysis
        - Combine static code analysis with dynamic system simulation
        - Static or dynamic model of inter-process cross-dependencies

Control/Data Flow Graph (CDFG)
Analytical System Evaluation

- Modular Performance Analysis (MPA)
  - Network calculus, real-time calculus (RTC)

Source: C. Haubelt, J. Teich, DATE '09 Tutorial

MPSoC Analysis with MPA

Source: C. Haubelt, J. Teich, DATE '09 Tutorial
Simulation

- Create stimuli and simulate model

  ![Simulation Diagram]

- **Inputs**
  - Specification
    - Used to create interesting stimuli and monitors (golden output)
  - Model of DUT
    - Typically written in HDL or C or both

- **Output**
  - Failed test vectors (validation)
  - Quality metrics (evaluation)

  ➢ **Speed vs. accuracy**
  ➢ Fundamental tradeoff

Hybrid System-Level Methods

- **Static timing back-annotation**
  - Analytical one-time estimation
  - Instructions, basic blocks or functions
  - Timing, energy, …

- **Dynamic system simulation**
  - System description language
  - Simulation host
  - Functionality & timing/power/…
  - Generate trace

- **Timing analysis**
  - Latency, throughput, response time, etc.

Source: C. Haubelt, J. Teich
Trace-Driven Simulation

- **Drive simulation via pre-existing, static traces**
  - Traces for system block behavior
  - Traces obtained from fast functional-only simulation

- **Examples**
  - Trace-driven simulation
  - Arrival curve extraction from traces
  - Trace generation from arrival curves

Source: C. Haubelt, J. Teich, DATE '09 Tutorial

Lecture 7: Summary

- **Source-level profiling**
  - Early pruning and exploration

- **Static analysis**
  - Worst/best/average case bounds
    - Tightness of bounds?
  - Execution time analysis of single task
  - Real-time calculus for concurrent systems
    - Max-plus algebra

- **Simulation**
  - Host-compiled system-level simulation
    - Fast and accurate, no guarantees