Weighted Control Scheduling

Aravind Vijayakumar and Forrest Brewer
UC Santa Barbara
Overview

- Optimal scheduling of forward branching CDFGs under weighted average latency
- BDD-based automata exploration
- Accommodates correlated branches
- Implicitly allows speculation, dynamic branch re-ordering, complex binding constraints

“Weighted Control Scheduling”, Aravind Vijayakumar and Forrest Brewer, ICCAD ’05
Motivation

Number Of Runs

Correlated branch profile

“Weighted Control Scheduling”, Aravind Vijayakumar and Forrest Brewer, ICCAD ’05
Requirements

- Fixed hardware resources
  - Arbitration

- Optimization across all paths
  - Control alternatives are *not* equal

- Accommodate 1000s to millions of paths
  - $K$ independent branches $\Rightarrow 2^K$ control paths
  - Need to find complete set of compatible (causal) control paths
Definition of Terms

- Control operation
- Control cube
- Control case (path)
  - Path in Red
  - Trace
  - Cost
- Ensemble schedule
  - Cost

“Weighted Control Scheduling”, Aravind Vijayakumar and Forrest Brewer, ICCAD ’05
Problem Formulation

- Weighted average latency metric
- Ensures **every** trace is part of **some** ensemble of **minimal cost**
- Entails new labeling and pruning mechanism
- Maintenance of potential trace latency
- Schedule termination criterion
- Added complexity allows solution to be performance driven

“Weighted Control Scheduling”, Aravind Vijayakumar and Forrest Brewer, ICCAD ’05
Weight Model, Restrictions

- Weights – positive, identified with disjoint control path sets
- Weight = Execution probability, in practice
- Forward branching structure
- Control operations binary

“Weighted Control Scheduling”, Aravind Vijayakumar and Forrest Brewer, ICCAD ’05
Bit Resolution

- Expected source of weights: simulation
- Finite precision of measurement
  - Measurement error
- Finite bit-width for cost sufficient
- True even with round-off
  - Absolute vs. Relative
Symbolic Scheduling - Overview

- Non-deterministic Finite Automata (NFA) based modeling of process
- Operation behavior modeled as small NFA
- BDD representation of composite transitions
- CDFG modeled by guarding operation transitions
- Solution consists of a (set of) trace(s) through the model
Operator Models

- NFA models encode operation I/O behavior
- Example:
  Simple 1 input, 1 cycle operation model

- State of operator enables transition of dependent operand.

“Weighted Control Scheduling”, Aravind Vijayakumar and Forrest Brewer, ICCAD ’05
Example Cache Model

- Simple to construct abstraction of signaling
- E.g. 2 (hit), 5 (miss) cycle latency cached memory access:

```
s0 ← s1 H → s2 H
  1

s0 ← s1 M → s2 M → s3 M → s4 M → s5 M
  1
```

“Weighted Control Scheduling”, Aravind Vijayakumar and Forrest Brewer, ICCAD ’05
Controls

- Control operations need to bifurcate model state space
- Example:
  1 cycle, 1 input, 1 bit control op:

- **Bold** bit distinguishes traces
if (c == T) {
    x++; // a
    x>>1; // b
} else {
    x<<1; // e
    x++; // f
}
Composite NFA Model of CDFG

- Boolean encoding of composite FSM

Composite State Encoding

\[
\begin{align*}
\text{c} & \quad \text{a} & \quad \text{b} & \quad \text{e} & \quad \text{f} \\
10 & 1 & 0 & 0 & 0 \quad (c=F, a) \\
0- & 1 \quad \text{Triggering for ‘b’} \\
11 & 1 \\
10 & - & - & - & 1 \quad \text{Join} \\
11 & - & 1 & - & - \quad \text{point}
\end{align*}
\]
Weighted example
Representative traces

```
\begin{tabular}{|c|c|c|c|c|}
\hline
\text{t=1} & c=T & c & \text{(8)} & \text{(3)} \\
\hline
\text{t=2} & c & e & \text{(2)} & \text{(12)} \\
\hline
\text{t=3} & c & a & \text{(12)} & \text{(3)} \\
\hline
\end{tabular}
```

"Weighted Control Scheduling", Aravind Vijayakumar and Forrest Brewer, ICCAD '05
Scheduling - I

- BFS of state space
- Start with initial operands known
- Forward image computation
Scheduling - II

- Bifurcating traces
- Speculation
Scheduling - III

- First terminals seen
- Labeling of costs
- Causality
Validation

- Complete but incompatible pair of traces

Traces
- c=T, cost = 8
- c=F, cost = 2

Conditional speculation

“Weighted Control Scheduling”, Aravind Vijayakumar and Forrest Brewer, ICCAD ’05
Cost representation

- Cost – normalized weight x latency
- Cost represented as binary-encoded integer
- BDD encoding attached to state BDD
- Example: (after 2 cycles)

```
c a b e f  <-cost->
11 1 1 0 0  01000  (4 x 2 = 8)
10 0 0 1 1  00010  (1 x 2 = 2)
```
Scheduling - IV

Complete solutions present!
Scheduling - V

- Reverse propagate labels

“Weighted Control Scheduling”, Aravind Vijayakumar and Forrest Brewer, ICCAD ’05
Scheduling - VI

- States may have multiple cost labels
Scheduling - VII

- Parallel addition across resolving transitions
Solution Space and Optimality

- Constant cost surface
- Earliest termination
- Extent of exploration

"Weighted Control Scheduling", Aravind Vijayakumar and Forrest Brewer, ICCAD ’05
Algorithm

Record minimum

Optimal?

Y

Halt

N

Valid?

Y

N

Weighted validation

Complete?

N

Add time step

Y

Record terminals

“Weighted Control Scheduling”, Aravind Vijayakumar and Forrest Brewer, ICCAD ’05
Note on Cost Calculation

- Addition of trace costs commutes over controls
- Not restricted to numbers, addition!
- Any set of labels
- Any binary operator that commutes over the controls
- Cheap BDD representation of operation needed
- For addition $\rightarrow$ linear with interleaved variables

“Weighted Control Scheduling”, Aravind Vijayakumar and Forrest Brewer, ICCAD ’05
Note on Termination Criterion

- Exploration cannot stop at state saturation!
  - Only true for restricted class of models
- Optimal cost
- Path saturation
- Some sequential constraints may render solution impossible
Some experiments

- Rotor – 28 ops, 3 controls, longest path 7
- ADPCM encoder – 45 ops, 11 controls, longest path 15
- Kim54 – 54 ops, 5 controls, longest path 9 (1 cycle multiply)
- Synthetic CDFG – 6 ops, 1 cached access, 1 black box operator with dynamic abort
- BDD package – CUDD
- P4 3GHz/2GB Linux PC
# Experimental results - Rotor

<table>
<thead>
<tr>
<th>Set</th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
</tr>
</thead>
<tbody>
<tr>
<td>W0</td>
<td>0.25</td>
<td>0.25</td>
<td>0.25</td>
<td>0.25</td>
</tr>
<tr>
<td>W1</td>
<td>0.97</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>W2</td>
<td>0.6</td>
<td>0.15</td>
<td>0.15</td>
<td>0.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Res.*</th>
<th>None</th>
<th>W0</th>
<th>W1</th>
<th>W2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1±,1×,1==</td>
<td>11</td>
<td>10.25(11)</td>
<td>9.05(12)</td>
<td>9.6(12)</td>
</tr>
<tr>
<td>2±,2×,1==</td>
<td>8</td>
<td>7.75(8)</td>
<td>7.03(8)</td>
<td>7.35(9)</td>
</tr>
<tr>
<td>2±,1×,2==</td>
<td>9</td>
<td>9(9)</td>
<td>8.04(10)</td>
<td>8.5(10)</td>
</tr>
</tbody>
</table>

*1 table look-up

“Weighted Control Scheduling”, Aravind Vijayakumar and Forrest Brewer, ICCAD '05
Experimental results – Kim54, ADPCM

- **Kim54**: $2^+, 2^-, 2^\times, 1^{==}$
  - Wavesched – 10/14/12.6
  - Optimal – 7/9/7.875
    (shortest/longest/average)
- **ADPCM encoder**: $1^{\pm}, 2^{==}, 2^{[]}, 1^{<<}$
- **Separately scheduled pre-loop, body, post-loop**
- **For 10 iterations**:
  - Spark – 192
  - Optimal – 157.75(164)
Experimental Results – Synthetic CDFG

- Black box subsystem imposes sequential constraint
- Cache may fail to meet timing

<table>
<thead>
<tr>
<th>Metric</th>
<th>Shortest</th>
<th>Longest</th>
<th>Expectation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Worst case</td>
<td>6</td>
<td>8</td>
<td>6.16</td>
</tr>
<tr>
<td>Weighted, Independent</td>
<td>5</td>
<td>11</td>
<td>5.48</td>
</tr>
<tr>
<td>Weighted, Correlated</td>
<td>5</td>
<td>11</td>
<td>5.45</td>
</tr>
</tbody>
</table>
Conclusions, Future Work

- Technique for optimizing weighted average latency
- No assumption of branch independence
- May be used to develop heuristics
- Explore other cost functions
- Basis for abstraction

“Weighted Control Scheduling”, Aravind Vijayakumar and Forrest Brewer, ICCAD ’05