A Template for Predictability Definitions with Supporting Evidence

Daniel Grund[1] Jan Reineke[2] Reinhard Wilhelm[1]

[1]Saarland University, Saarbrücken, Germany

[2]University of California, Berkeley, USA

Workshop on
Predictability and Performance in Embedded Systems

Outline

Outline

Motivation

The Platitude

Complexity of embedded systems increases.
They become more and more “unpredictable.”

- Timing analysis
  - current methods will not scale to future systems
  - need better analyses or more predictable systems
- Projects
  - PREDATOR: design for predictability and efficiency
  - PRET: reintroduce timing predictability and repeatability
  - MERASA: guarantee analyzability and predictability

The Problem

- We don't know what predictability actually is or should be
- People mean different things when saying "predictability"
- Criteria for predictability are mostly
  - intuitive
  - case-based
  - phenomenological, symptom-based

Examples

Predictability	better	worse
CPU pipeline	in order	out of order
branch prediction	static	dynamic
cache replacement	LRU	FIFO
scheduling	static	dynamic preemptive
arbitration	TDMA	FCFS

- This is mostly based on intuition
- Is there a common underlying principle?

Vision

- Derive formal definition of predictability
- In the ideal case
  - uniformly applicable
  - rather constructive than existential

Outline

What is Predictability?

SAARLAND UNIVERSITY
COMPUTER SCIENCE

Oxford Dictionary

predictable	*adjective, able to be predicted*
to predict	*verb, state that a specified event will happen in the future*

- Want to consider hardware/software systems
- That is, deterministic and finite systems
- Concretize above definition for such systems

Aspect 1: Property to be Predicted

- System behaviors can be described by a set of traces
- Exact behavior often irrelevant
- Derived properties are interesting

$\Rightarrow$ Which property of the system to predict?

  - number of certain events on a trace
  - maximal length of traces (WCET)

Aspect 2: Sources of Uncertainty

- Deterministic finite systems
- Any property can be determined exactly

- However, properties depend on something unknown
- Predictions shall hold in any case

$\Rightarrow$ What are the sources of uncertainty?
  - program input $\rightarrow$ execution time

Aspect 3: Quality Measure

- Generally, predictability is not a Boolean property
- Allow shades of gray
- How well can a property be predicted?

$\Rightarrow$ What is the quality measure on predictions?

Aspect 4: Inherence

- Consider
  - two systems S and T
  - an analysis A for a system property P
- Assume A can determine P for S better than for T
- Should S therefore be more predictable than T?
- We say no; there could be A' where it is vice versa

$\Rightarrow$ Predictability should be a property inherent to the system

Predictability Template

Proposition

The notion of predictability should capture if,
and to what level of precision,
a specified property of a system
can be predicted by an optimal analysis.

- In particular, a definition should state
  - Property to determine
  - Sources of uncertainty
  - Quality measure

Example: Timing Predictability

Definition (Timing Predictability)

$$\Pr_p(\mathcal{Q}, \mathcal{I}) := \min_{q_1, q_2 \in \mathcal{Q}} \min_{i_1, i_2 \in \mathcal{I}} \frac{T_p(q_1, i_1)}{T_p(q_2, i_2)}$$

- Property: execution time of a program
- Sources of uncertainty: program input & hardware state
- Quality measure: variance in execution time

- Measure is within [0..1]
- 1 means perfectly predictable

Outline

Supporting Evidence?

- Work on architectural components that are better predictable
- Usually based on sensible, yet informal intuitions
- Try to cast that work in terms of our template

Examples — Part 1

- Lickly et al.: Pred. programming on a precision timed arch. (PRET)
- Bhat & Mueller: Making DRAM refresh predictable
- Barre et al.: A predictable SMT scheme for hard real-time

Hardware unit	Property	Source of uncertainty	Quality measure
Thread-interleaved pipeline and scratchpad memories	Execution time	Uncertainty about initial state and execution context	Variability in execution times
DRAM controller	Latency of DRAM accesses	Occurrence of refreshes	Variability in latencies
SMT processor	Execution time of tasks in real-time thread	Uncertainty about execution context, i.e., other tasks executing in non-real-time threads	Variability in execution times

Examples — Part 2

- Bodin & Puaut: A WCET-oriented static branch prediction scheme
- Schoeberl et al.: Towards time-pred. data caches for chip-MP
- Rochange & Sainrat: A time-pred. exec. mode for superscalar pipelines

Hardware unit	Property	Source of uncertainty	Quality measure
Branch predictor	Number of branch mispredictions	Analysis imprecision (Uncertainty about initial predictor state)	Statically computed bound (Variability in mispredictions)
Memory hierarchy	Number of data cache hits	(Percentage of accesses that can be statically classified)	Among others, uncertainty about addresses of data accesses
Superscalar out-of-order pipeline	Execution time of basic blocks	Analysis imprecision (Uncertainty about the pipeline state at basic block boundaries)	Qualitative: analysis practically feasible (Variability in execution times of basic blocks)

Discussion

SAARLAND UNIVERSITY
COMPUTER SCIENCE

- 13 works on predictability improvements tabulated in paper
- Most fit well into the predictability template
- Parameters are indeed instantiated differently
- Many approaches disagree on quality measure
  - evaluation of predictability improvement using analyses
  - practical approach
  - not conclusive on (inherent) predictability

Outline

Summary

- Key aspects of a predictability definition:
  - Property to be predicted
  - Sources of uncertainty
  - Quality measure
  - Inherence
- Predictability template provides mental frame
- Most works on predictability can be cast as template instance