Making Real-Time Systems Time-Predictable

Peter PUSCHNER
peter.puschner@tuwien.ac.at
Contents

Real-time systems
WCET analysis
Approaching time predictability
Real-Time System

Timing constraints on operations
  (e.g., deadline from event to completion of response)

Hard real-time system:

- Timing constraints must hold under all circumstances
  (even under peak load)
- Failures may have severe consequences
  e.g., nuclear power station, medical equipment, airbag

Note: real-time computing $\neq$ fast computing
Task set with precedence constraints and deadline

Task sequence: execution times, response time

Response time < deadline ???
Timing-Analysis Issues

Schedulability, Scheduling

- Task set, including WCETs
- Precedence relations
- Synchronization, communication, mutual exclusion
- Priorities, ...

WCET-analysis

- Simple tasks

Interference ... (nasty and therefore widely neglected)
- “external” changes of task state that influence exec. time
Worst-Case Execution Time

Def. Worst Case Execution Time (WCET):
WCET of software is the time it takes to execute
• a given piece of code
• in a given application context (inputs, state)
• on a given machine

Cave: measurements are often unsafe ⇔ complexity
Cave: execution time is not response time
WCET Determinants

Possible sequences of actions of the task (= execution paths)
⇔ code, context, machine

The duration of each occurrence of an action on each possible (= feasible) path
⇔ machine, HW state
WCET Calculation as Optimization

Program

$t_i$ ... exec. time of edge $i$

$x_i$ ... exec. frequency of edge $i$

total execution time:

$maximize \ \sum x_i \cdot t_i$

WCET
WCET Analysis Problems

Data and state dependent **hardware timing**
Different execution paths for different inputs

**Infeasible paths**

- Static analysis: highly complex models needed; Simplifications cause pessimism
- Anomalies: obstacle to compositional timing analysis
Timing Anomaly

out-of-order pipeline + cache + data dependencies:
Vision

Software/hardware architecture provides predictable and composable timing
We need ...

Regularity of shape:
→ dimensions are easy to assess, describe

Composability: it has the same dimensions under all circumstances (stand alone, when integrated, ...)

Compositionality: combination function is simple
Goal-oriented Approach

Goal-oriented approach towards temp. predictability:
Control sequencing of all actions instead of being controlled by the environment (data, interrupts)

- Generate **single-path code**:
  - Code without input-data dependent branches
  - Use of predicated execution (possibly with speculation)
Single-Path Transformation

Transform input-data dependent branches into sequential predicated code, rest remains unchanged

⇒ Technique based on: if-conversion
Programming for the Worst Case

Try to produce code that is free from input-data dependent control decisions

Restrict operations that are only executed for a subset of the input-data space to a minimum
Time-predictable Prefetching
Time-predictable Prefetching (2)
Topics

- WCET analysis & Compilation
- Programming and Code generation
- Time-predictable computer architectures
  - Instruction Filter for ARM
  - Vector Co-Processor for RISC-V
- OS support for predictability and composability
- Time-predictability for COTS hardware
- Timing and Security