

Semi-Formal Verification of Embedded Linux Systems Using Trace-Based Models

Benno Bielmeier¹
Wolfgang Maurer^{1,2}

¹Technical University of Applied Sciences Regensburg ²Siemens AG, Corporate Research

December 06, 2022

About: BB [ビエルマイヤー]

- ▶ Technical University of Applied Sciences Regensburg
 - ▶ Research Master Student
- ▶ Tokyo University of Sciences (東京理科大学)
 - ▶ Research Internship



About: WM [マウラー]

- ▶ Siemens Technology:
Corporate Competence Centre Embedded Linux
 - ▶ Civil Infrastructure Platform
- ▶ Technical University of Applied Sciences Regensburg
 - ▶ Theoretical Computer Science
 - ▶ Head of Digitalisation 研究室



About: BB [ビエルマイヤー]

- ▶ Technical University of Applied Sciences Regensburg
 - ▶ Research Master Student
- ▶ Tokyo University of Sciences (東京理科大学)
 - ▶ Research Internship









What?

- ▶ Safety + Security
- ▶ Real-Time
- ▶ Sustainability

Why?

- ▶ Features
- ▶ Cost
- ▶ Time

Need for Mixed Criticality

- ▶ COTS hardware
- ▶ Software-Intensive control systems
- ▶ Convenience features + determinism

OSS Challenges

- ▶ Self-organised, informal development process
- ▶ High complexity
- ▶ Rapid change

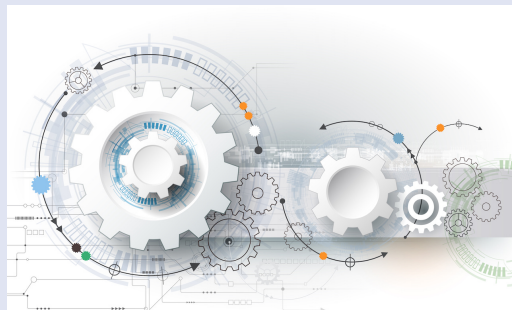
Beyond Linux

- ▶ RTOS kernels, hypervisor, middleware, ...

Process (People)



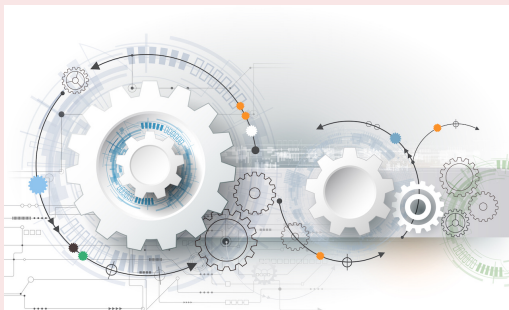
Technology (Artefacts)



Process (People)



Technology (Artefacts)



“Assessing whether a system is safe requires understanding the system sufficiently”

Speakers

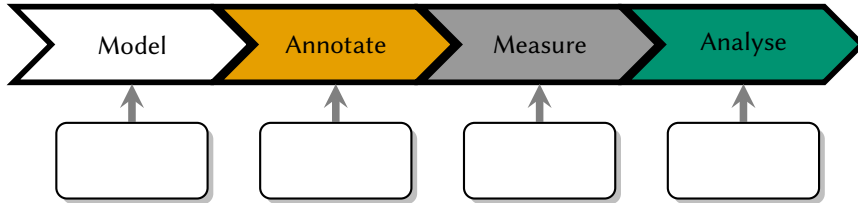


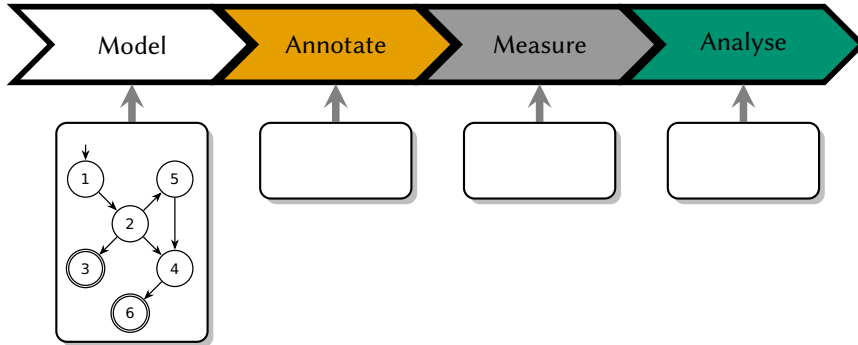
Philipp Ahmann

Product Manager, Robert Bosch GmbH

Philipp Ahmann is a technical business development manager at Robert Bosch GmbH with focus on Open Source activities. He represents the ELISA project of the Linux Foundation as technical steering committee chair and leads the automotive as well as systems work group. He has more than... [Read More →](#)

→ Mon, Dec. 5, 11:00@OSSJ





Model Properties

- ▶ **Generality**
 - ▶ Applicability to multiple real-world situations
- ▶ **Realism**
 - ▶ Accurately represent real-world phenomena
- ▶ **Precision**
 - ▶ Minimise errors compared to real-world results

Model Properties

- ▶ Generality ✓
 - ▶ Applicability to multiple real-world situations
- ▶ Realism ✗
 - ▶ Accurately represent real-world phenomena
- ▶ Precision ✓
 - ▶ Minimise errors compared to real-world results

Established (Industry)

1. Measure (suitable load)
2. Determine Maximum



Image source: osdal.org

Established (Industry)

1. Measure (suitable load)
2. Determine Maximum

Established (Academic)

1. Count possible paths through code
2. Determine Maximum



Image source: osdal.org

Established (Industry)

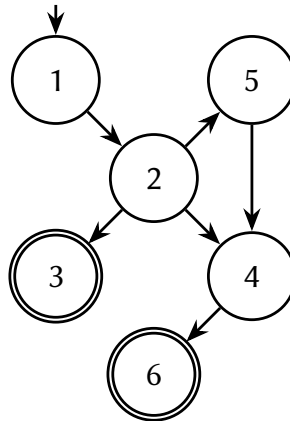
1. Measure (suitable load)
2. Determine Maximum

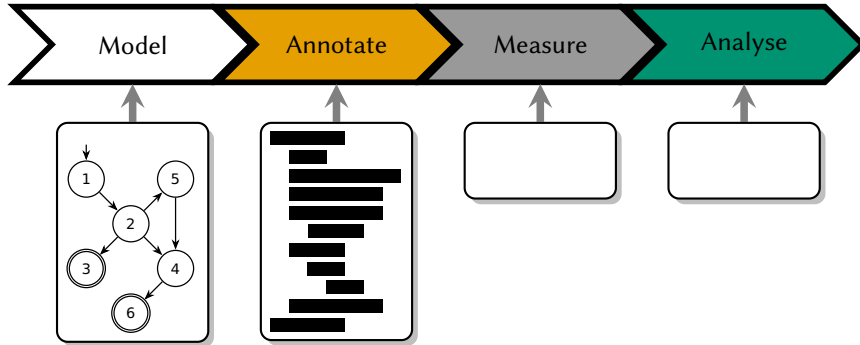
Established (Academic)

1. Count possible paths through code
2. Determine Maximum

Model-Based Approach

1. Provide *simplified* model
2. Instrument system, annotate model
3. Infer WCET from model





Requirements

- ▶ Record events with precise timing info
- ▶ Total order for recorded events possible

Requirements

- ▶ Record events with precise timing info
- ▶ Total order for recorded events possible

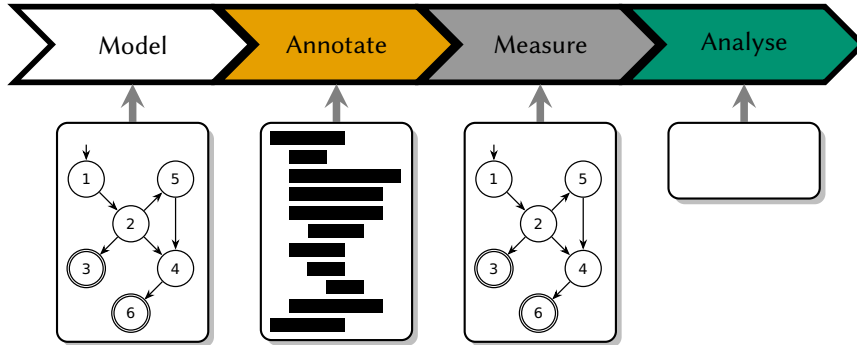
Kernel

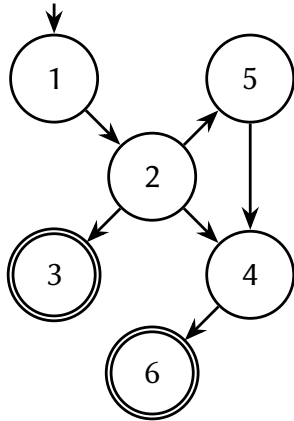
- ▶ Tracepoints + in-kernel handler
- ▶ Tracepoints + tracefs (ftrace)
- ▶ Custom probes and in-kernel handler

Userspace

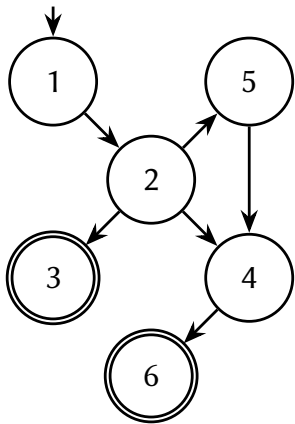
- ▶ Uprobe
- ▶ user_events
- ▶ User Statically-Defined Tracing (USDT)
- ▶ LTTng

Tracer	Domain	Static	System-wide	Mainline	Usability Measure	Analysis
TP+handler	K	✓	✗	✗		
TP+ftrace	K	✓	✓	✓/✗		
CP+handler	K	✓	✗	✗		
kprobes	K	✗	✓	✓		
Uprobe	U	✗	✓	✓		
USDT	U	✗	✓	(✗)		
user_event	U	✗	✓	(✓)		
LTTng	K+U	✓	✓			
eBPF	K+U	✗	✓	✓		
⋮	⋮	⋮	⋮	⋮	⋮	⋮





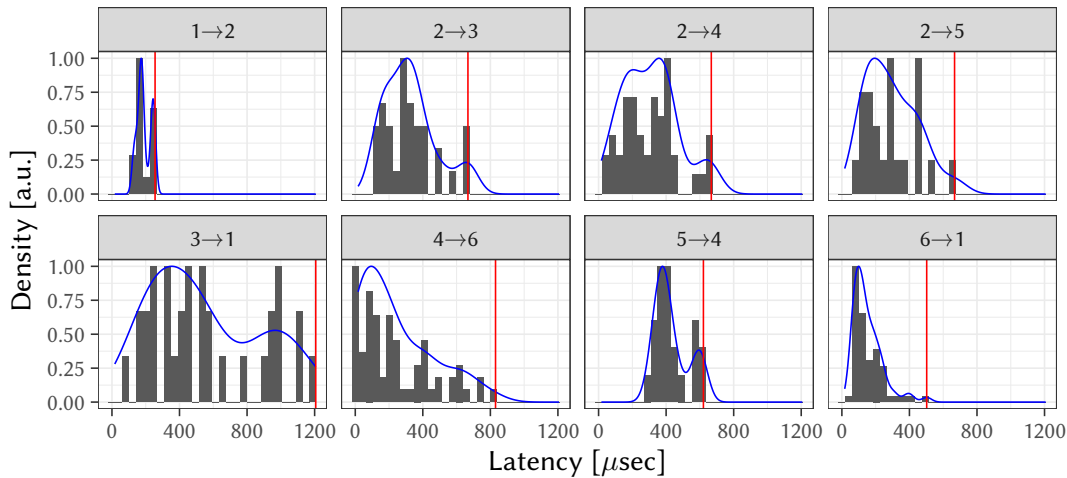
Represent *essentials* of a scenario!



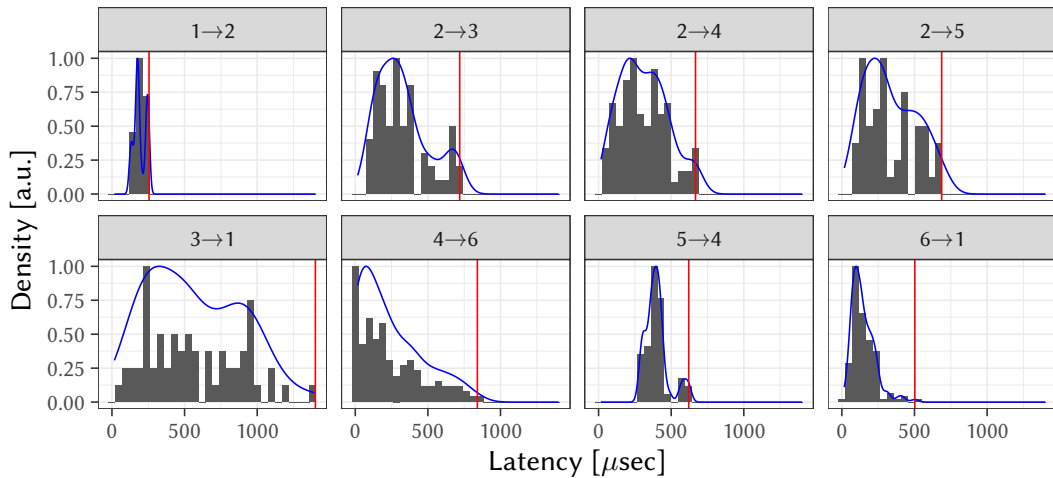
Transition	Duration
$1 \rightarrow 2$	2
$2 \rightarrow 3$	7
$1 \rightarrow 2$	1
$2 \rightarrow 4$	19
$\vdots$	$\vdots$

Represent *essentials* of a scenario!

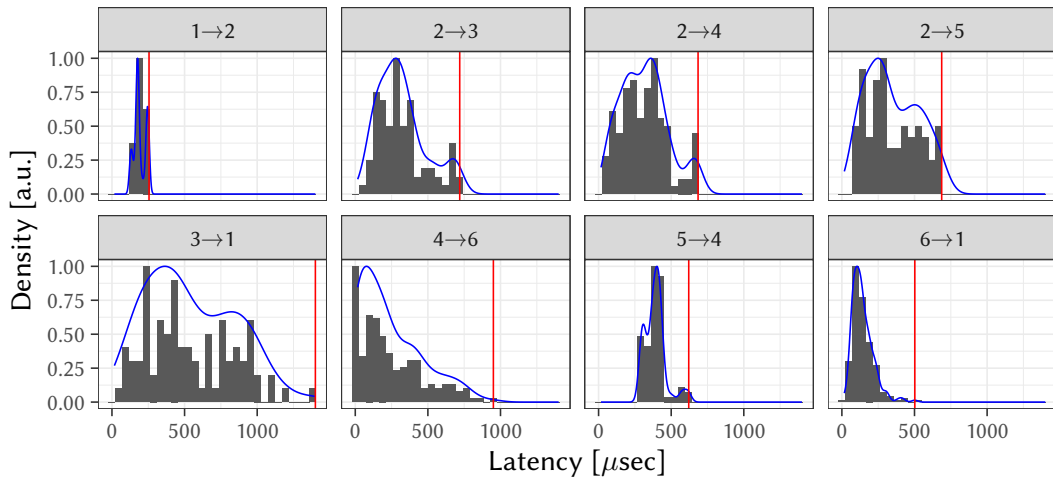
Runs = 100, State Changes = 388



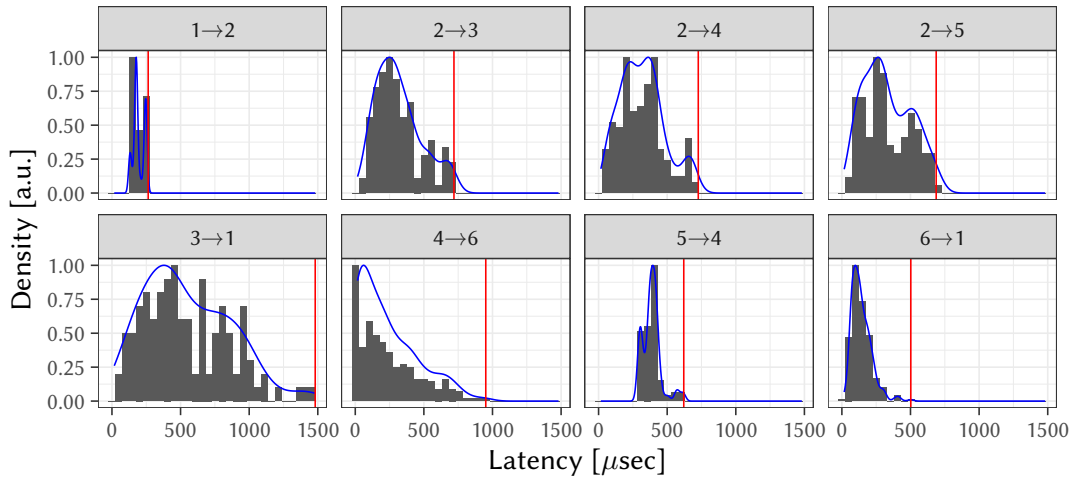
Runs = 200, State Changes = 784



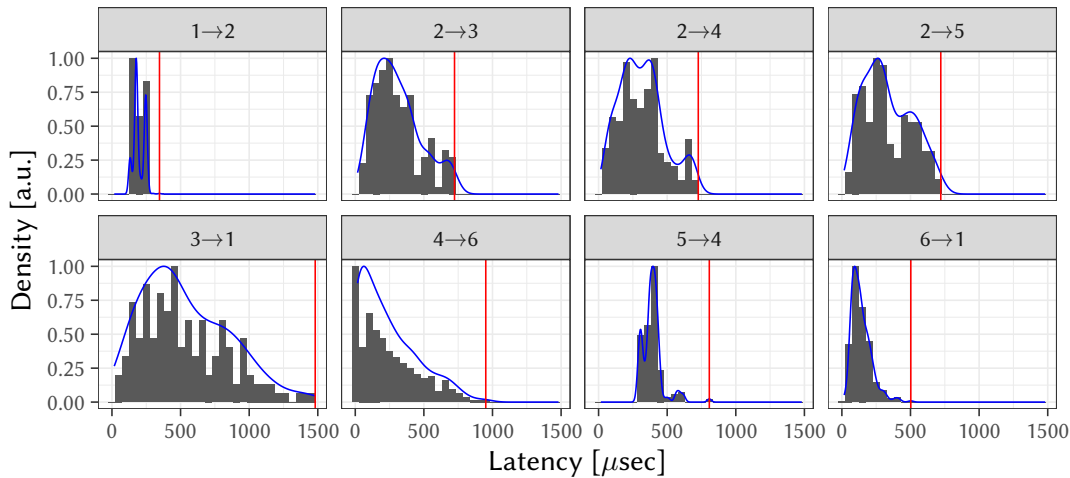
Runs = 300, State Changes = 1 190



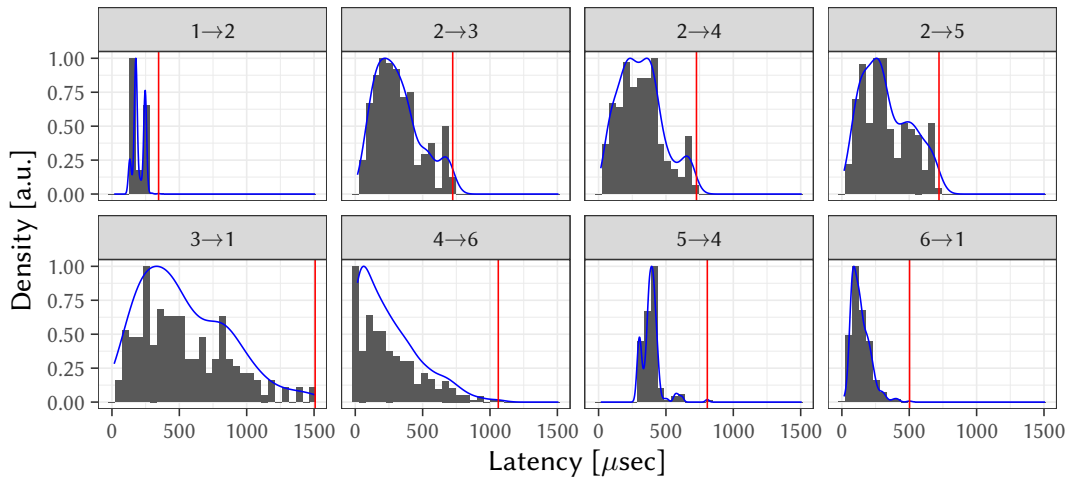
Runs = 400, State Changes = 1 586



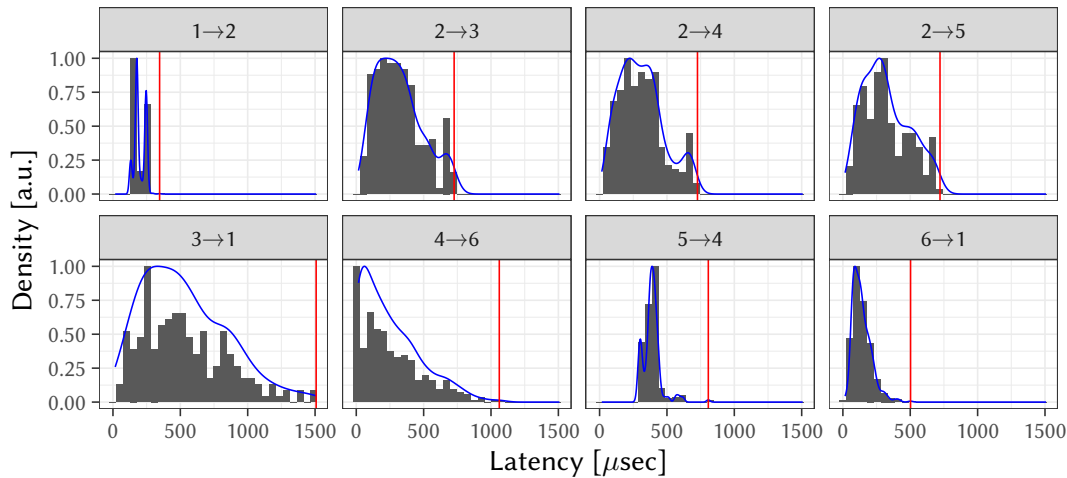
Runs = 500, State Changes = 1 972



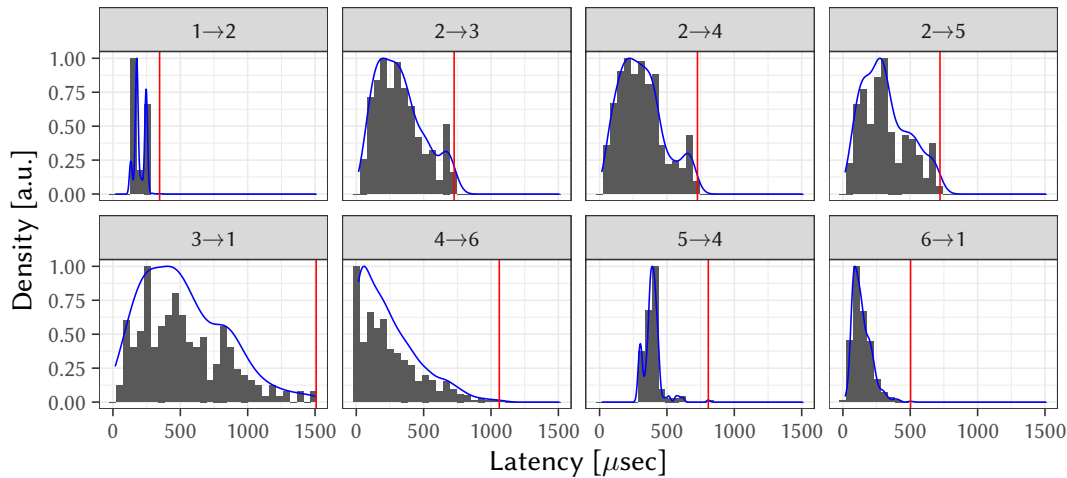
Runs = 600, State Changes = 2 378



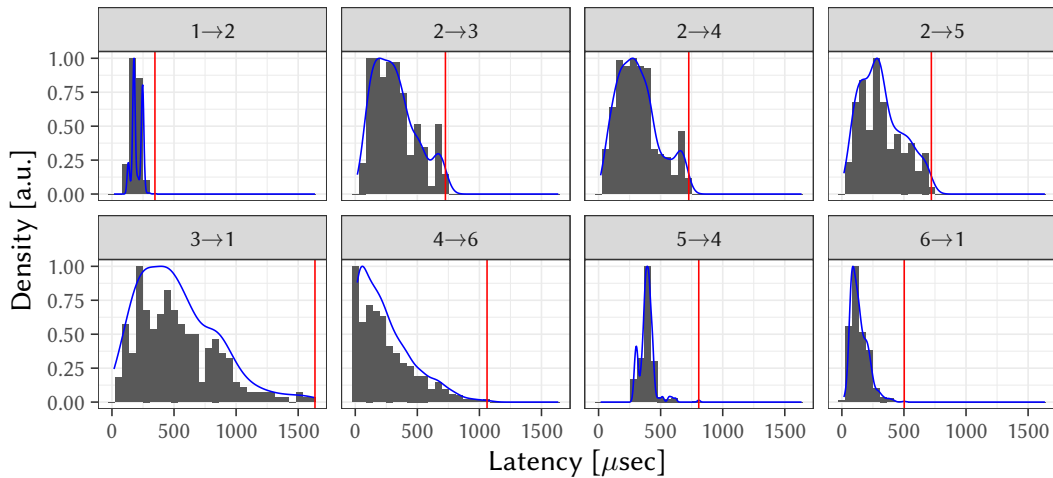
Runs = 700, State Changes = 2 773



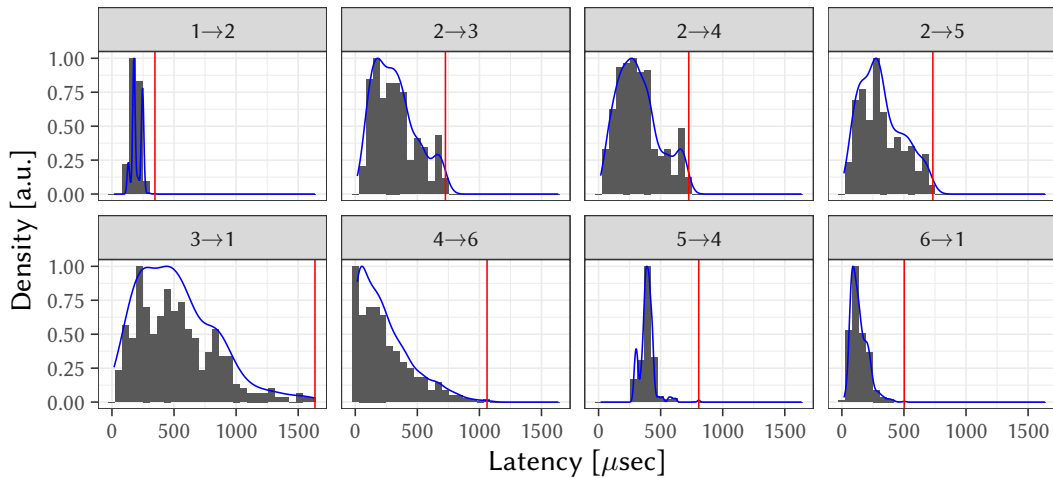
Runs = 800, State Changes = 3 174

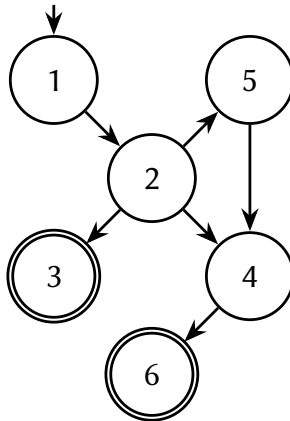


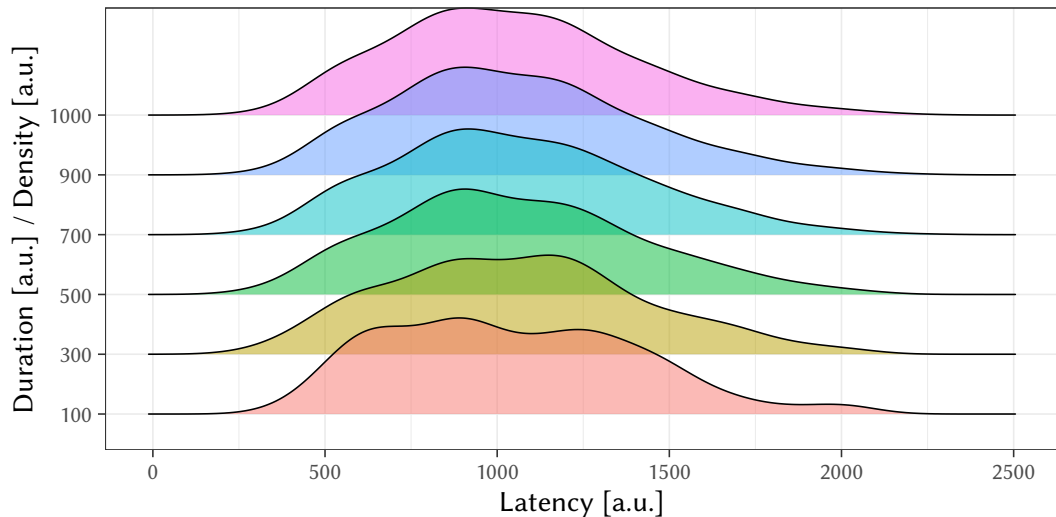
Runs = 900, State Changes = 3 575

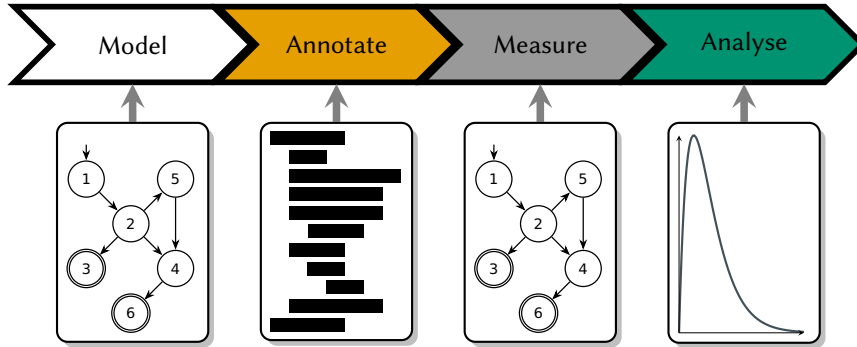


Runs = 1 000, State Changes = 3 969



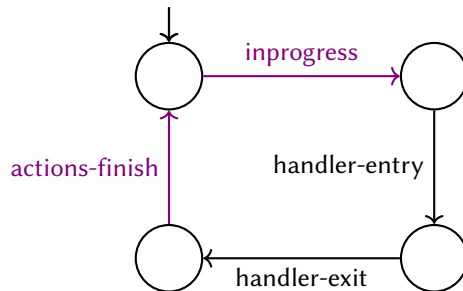


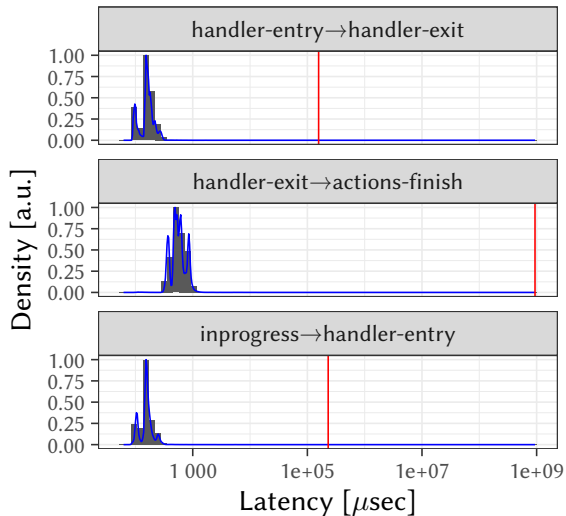


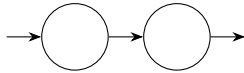


IRQ Handling

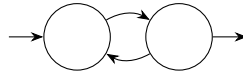
- ▶ Base pattern for RT loop
- ▶ Conceptual:
 - ▶ Receive
 - ▶ Process
 - ▶ React



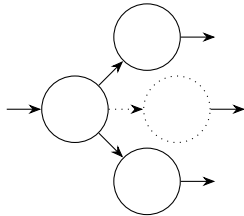




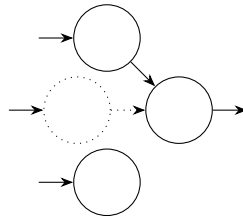
(a) Sequence



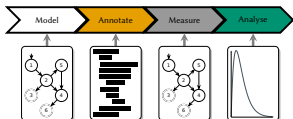
(b) Loop



(c) Fork



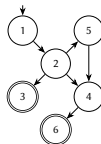
(d) Join



B. Bielmeier, W. Maurer

S.-Formal Verification w/ Trace-Based Models

December 06, 2022 19 / 26



Transition	Duration
1 → 2	2
2 → 3	7
1 → 2	1
2 → 4	19
⋮	⋮

Represent essentials of a scenario!

B. Bielmeier, W. Maurer

S.-Formal Verification w/ Trace-Based Models

December 06, 2022 15 / 26

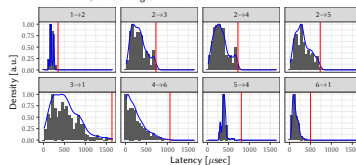
Tracer	Domain	Static	System-wide	Mainline	Usability Measure	Analysis
TP-handler	K	✓	✗	✗		
TP+trace	K	✓	✗	✗		
CP-handler	K	✓	✗	✗		
kprobes	K	✗	✓	✓		
Uprobe	U	✗	✓	✓		
USDT	U	✗	✓	(✗)		
user_event	U	✗	✓	(✓)		
LTTrg	K+U	✓	✓	✓		
ebPF	K+U	✗	✓	✓		
⋮	⋮	⋮	⋮	⋮	⋮	⋮

B. Bielmeier, W. Maurer

S.-Formal Verification w/ Trace-Based Models

December 06, 2022 13 / 26

Runs = 1 000, State Changes = 3 969

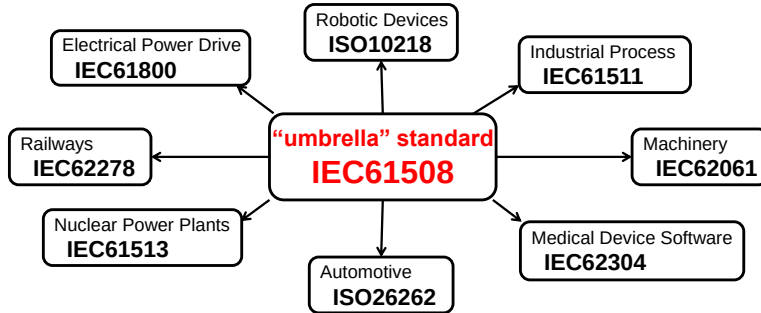


B. Bielmeier, W. Maurer

S.-Formal Verification w/ Trace-Based Models

December 06, 2022 16 / 26

Backup



Routes to Safety

- ▶ Standard compliant development
- ▶ Proven in use
- ▶ Compliant non-compliant development

```
--- a/include/trace/events/irq.h
+++ b/include/trace/events/irq.h
@@ -134,6 +134,20 @@
↪ TRACE_EVENT(irq_inprogress,
    TP_printk("irq=%d", __entry->irq)
);

+TRACE_EVENT(irq_finish_actions,
+  TP_PROTO(int irq, int ret),
+  TP_ARGS(irq, ret),
+  TP_STRUCT__entry(
+    __field( int, irq )
+    __field( int, ret )
+  ),
+  TP_fast_assign(
+    __entry->irq = irq;
+    __entry->ret = ret;
+  ),
+  TP_printk("irq=%d ret=%d", __entry->irq,
↪   __entry->ret)
+);
+
DECLARE_EVENT_CLASS(softirq,

    TP_PROTO(unsigned int vec_nr),
```

```
--- a/kernel/irq/handle.c
+++ b/kernel/irq/handle.c
@@ -182,6 +182,7 @@ irqreturn_t
↪ __handle_irq_event_percpu(struct
↪   irq_desc *desc)

    retval |= res;
}
+ trace_irq_finish_actions(irq, retval);

return retval;
}
```