WoCA: Avoiding Intermittent Execution in Embedded Systems by Worst-Case Analyses with Device States

June 24th, 2024

Phillip Raffeck, Johannes Maier, Peter Wägemann

Friedrich-Alexander-Universität Erlangen-Nürnberg

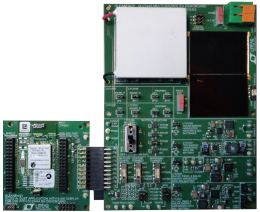
Project no 502947440 (Watwa)
Project no 502615015 (ResPECT)

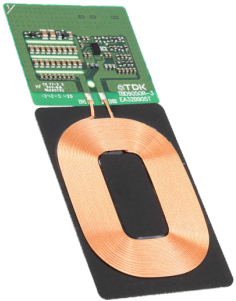
Device-Bound Energy-Constrained Systems

de Winkel
MobiSys
2022

Device-Bound Energy-Constrained Systems

de Winkel
MobiSys
2022



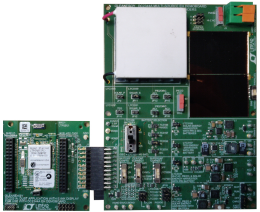


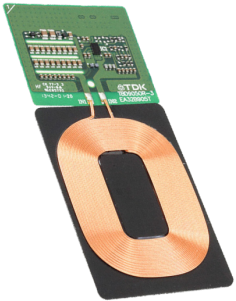
Energy Harvesting

- Solar
- Radio frequencies
- Piezo-/thermo-electric

Device-Bound Energy-Constrained Systems

de Winkel
MobiSys
2022





Energy Harvesting

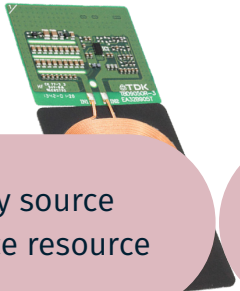
- Solar
- Radio frequencies
- Piezo-/thermo-electric

Device-Bound Systems

- Embedded microcontroller
- Sensors
- Wireless communication

Device-Bound Energy-Constrained Systems

de Winkel
MobiSys
2022



- Unreliable energy source
- Energy as a scarce resource

- Frequent device use
- Varying power demand

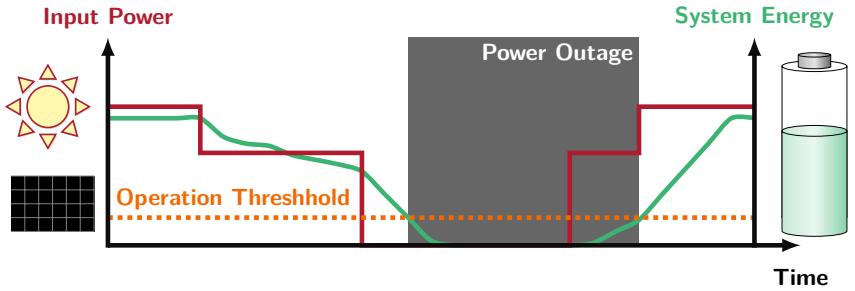
Energy Harvesting

- Solar
- Radio frequencies
- Piezo-/thermo-electric

Device-Bound Systems

- Embedded microcontroller
- Sensors
- Wireless communication

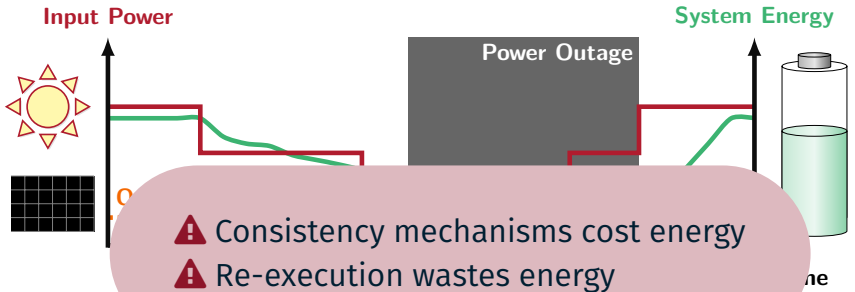
Intermittent Execution



Power Outages in Energy-Harvesting Systems

- Maintain **consistency** in the system through power outages
- Resume operation when sufficient **energy available**
- System state **checkpointing** in non-volatile memory
- Potentially **re-execute** (partial) device operations

Intermittent Execution



Power Outages in

- Maintain **consistency** in the system through power outages
- Resume operation when sufficient **energy available**
- System state **checkpointing** in non-volatile memory
- Potentially **re-execute** (partial) device operations

Example Taskset

Task τ_1

```
p = next_packet();
tx_on(HIGH_POWER);
tx_send(p);
tx_wait_for_done();
tx_off();
return;
```

Example Taskset

Task τ_1

```
p = next_packet();
tx_on(HIGH_POWER);
tx_send(p);
tx_wait_for_done();
tx_off();
return;
```

⚠ Device configuration determines power demand

Example Taskset

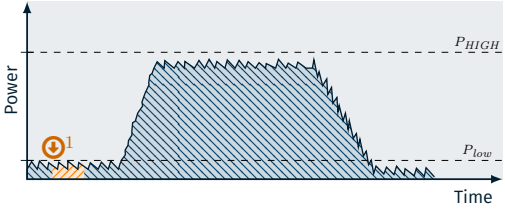
Task τ_1

```
p = next_packet();
tx_on(HIGH_POWER);
tx_send(p);
tx_wait_for_done();
tx_off();
return;
```

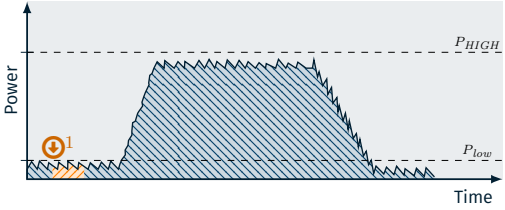
Interrupt ISR_1

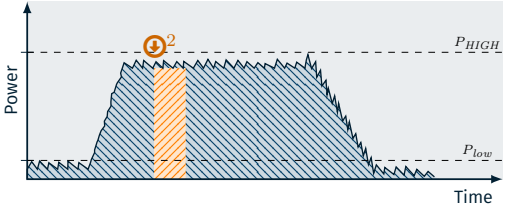
```
d = data_get();
buffer_add(d);
```

Whole-System Perspective

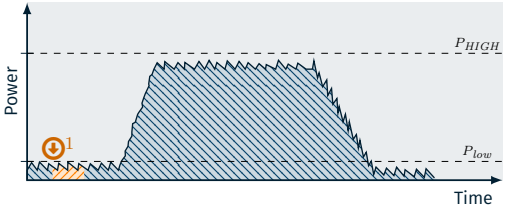


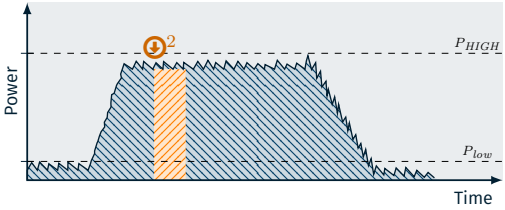
Whole-System Perspective





Whole-System Perspective





System context determines power demand

Problem Overview

- Problem of **device states** with **varying power demand**
- Problem of **asynchronous** device use in **different contexts**
- Problem of **re-execution** due to transactional operation semantics

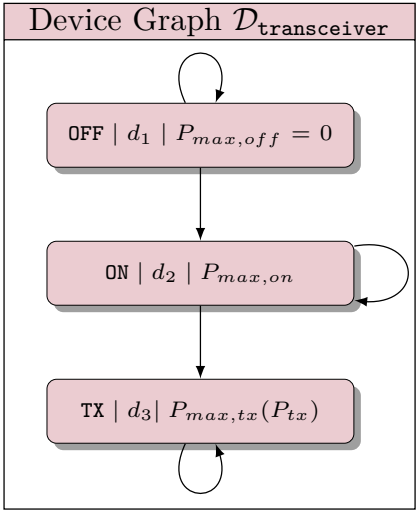
Problem Overview

- Problem of **device states** with **varying power demand**
- Problem of **asynchronous** device use in **different contexts**
- Problem of **re-execution** due to transactional operation semantics

Goals

- Forward-progress guarantees
- Consistency guarantees
- Efficient use of available energy

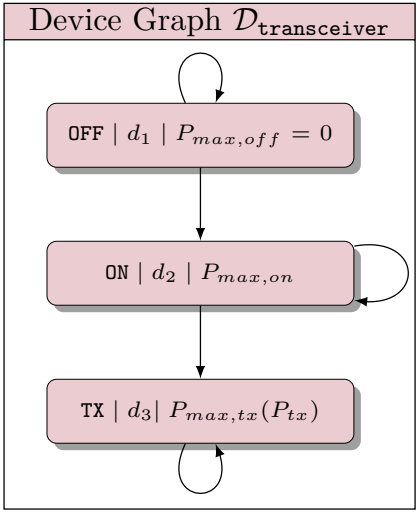
Modeling Device Behavior



Device Graph

- Device states

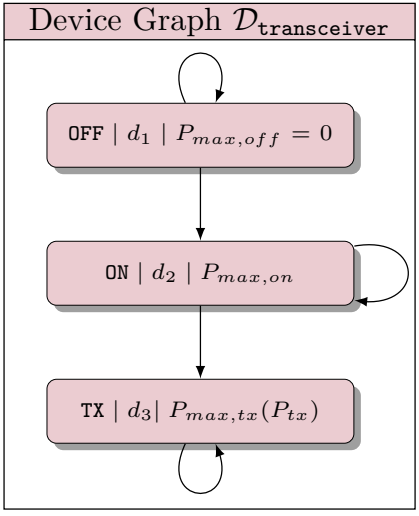
Modeling Device Behavior



Device Graph

- Device states
- Maximum power consumption of states

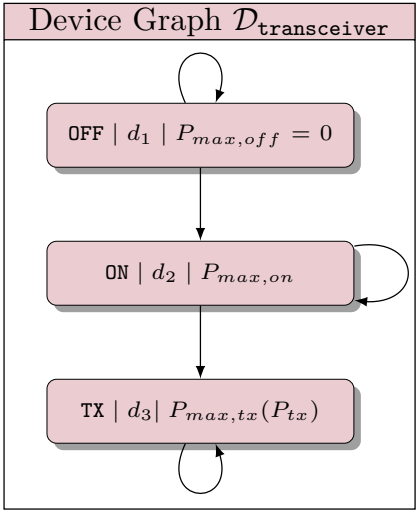
Modeling Device Behavior



Device Graph

- Device states
- Maximum power consumption of states
- Transitions between states

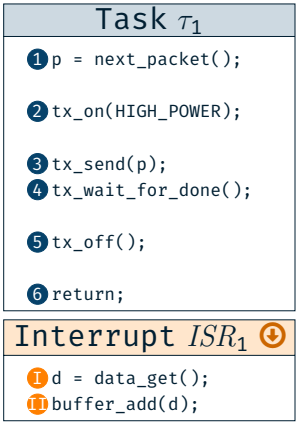
Modeling Device Behavior



Device Graph

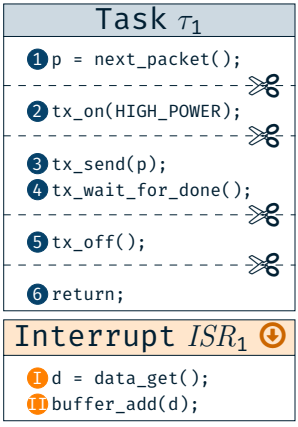
- Device states
- Maximum power consumption of states
- Transitions between states
- Internal device configuration
  - Output power
  - Bandwidth

Modeling System Behavior



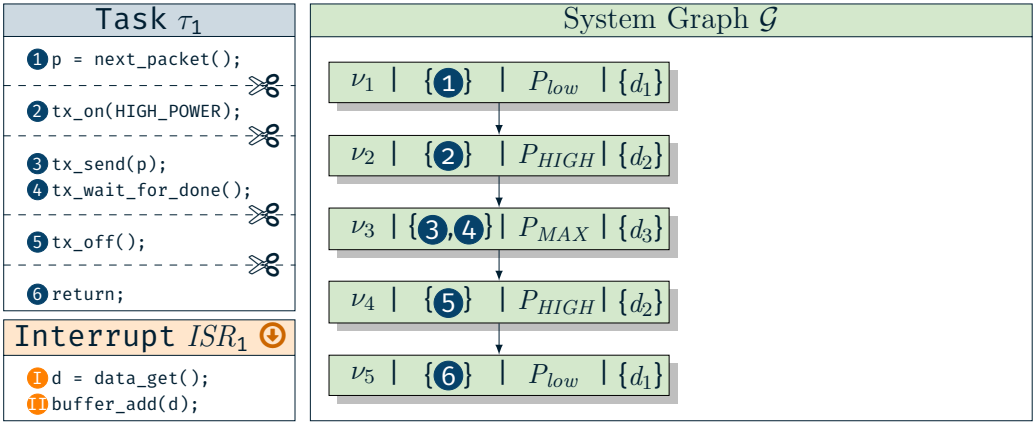
- Identify device-state changes on all system paths
- Decompose system into states with constant power consumption

Modeling System Behavior



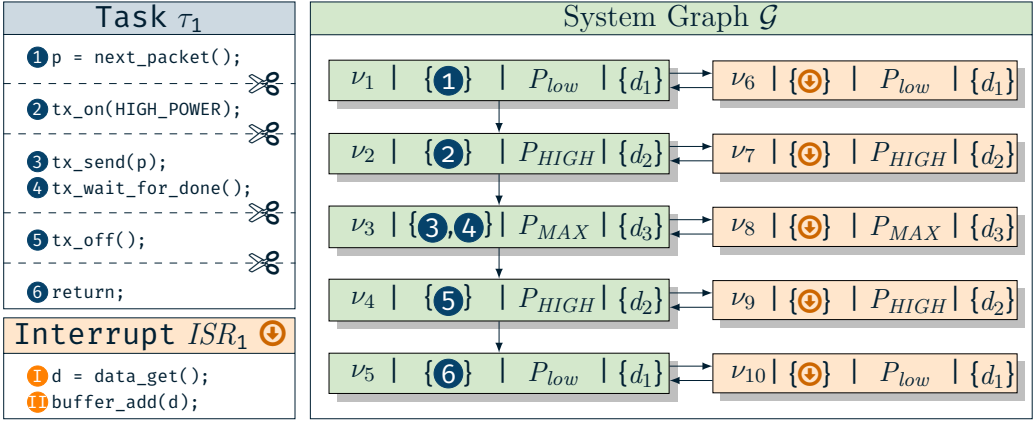
- Identify device-state changes on all system paths
- Decompose system into states with constant power consumption

Modeling System Behavior



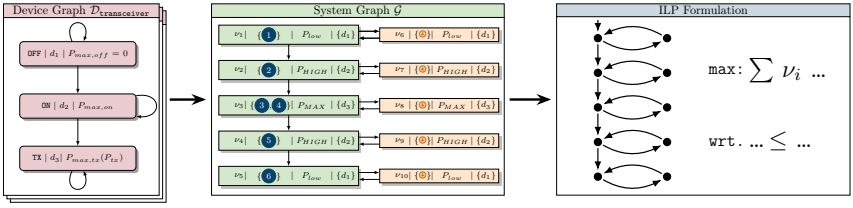
- Identify device-state changes on all system paths
- Decompose system into states with constant power consumption

Modeling System Behavior



- Identify device-state changes on all system paths
- Decompose system into states with constant power consumption

Guarantees through Analysis



- Convert system graph to optimization problem
- Find whole-system worst-case energy consumption of tasks
- Dimension energy storage accordingly
- Start tasks only if sufficient energy is present
- **Guaranteed transactional execution semantics**

From Code to Analysis to Bounds

Employed Toolchain

- Compilation with modified Clang/LLVM
  - Input: annotated source code and **device graphs**
  - Output: **system graph**, control-flow information, executable
- Resource-bound analysis with Platin Hepp, KPS, 2015 Maroun, WCET, 2024
  - Input: **system graph**, control-flow information, executable
  - Output: **energy-consumption bounds**

Evaluation

Evaluation Setup

Benchmarks

- `bsort`: computation only
- `temp`: uncomplicated device use (temperature-sensor readout)
- `send`: complex device use (LoRa transmission)
- `send-{PL,BW,SF}`: varying **p**ay**l**oad, **b**and**w**idth, **s**preading **f**actor
- `sca`: sensing, computation, actuation
- `sca-isr-{lf,hf}`: `sca` with low- and high-frequency interrupts

Evaluation Setup

Benchmarks

- `bsort`: computation only
- `temp`: uncomplicated device use (temperature-sensor readout)
- `send`: complex device use (LoRa transmission)
- `send-{PL,BW,SF}`: varying **p**ay**l**oad, **b**and**w**idth, **s**preading **f**actor
- `sca`: sensing, computation, actuation
- `sca-isr-{lf,hf}`: `sca` with low- and high-frequency interrupts

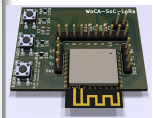
Approaches

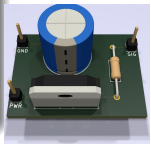
- WoCA
- no-ctx: device-aware, context-agnostic approach
- all-on: device-agnostic approach
- JIT-based: checkpoints on energy interrupt and before device use

Evaluation Setup

Benchmarks

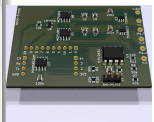
- `bsort`: computation only
- `temp`: uncomplicated device use (temperature-sensor readout)
- `send`: complex device use (LoRa transmission)
- `send-{PL,BW,SF}`: varying **p**ay**l**oad, **b**and**w**idth, **s**preading **f**actor
- `sca`: sensing, computation, actuation
- `sca-isr-{lf,hf}`: `sca` with low- and high-frequency interrupts



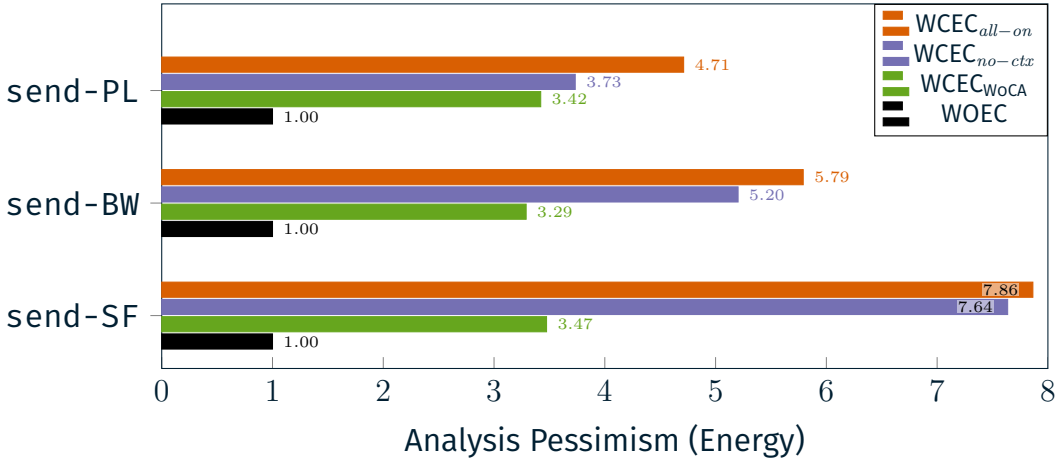


Approaches

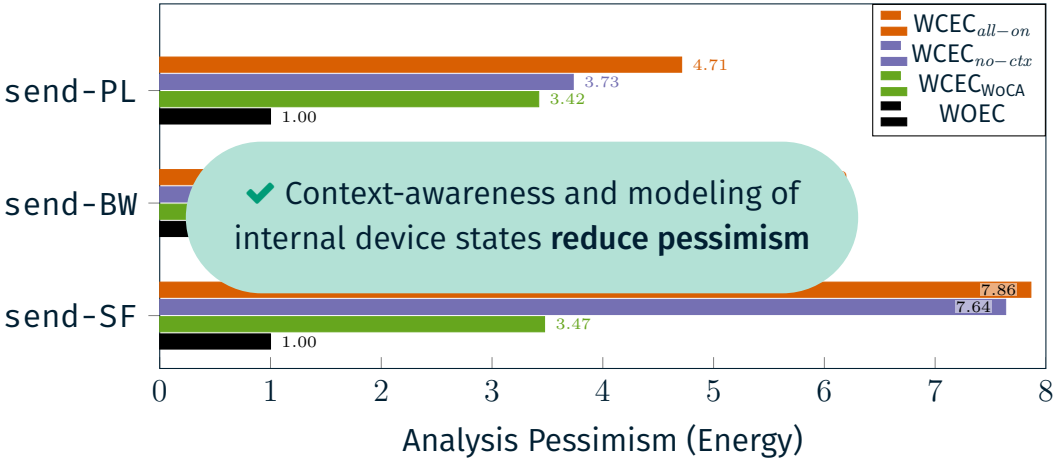
- WoCA
- no-ctx: device-aware, context-agnostic approach
- all-on: device-agnostic approach
- JIT-based: checkpoints on energy interrupt and before device use



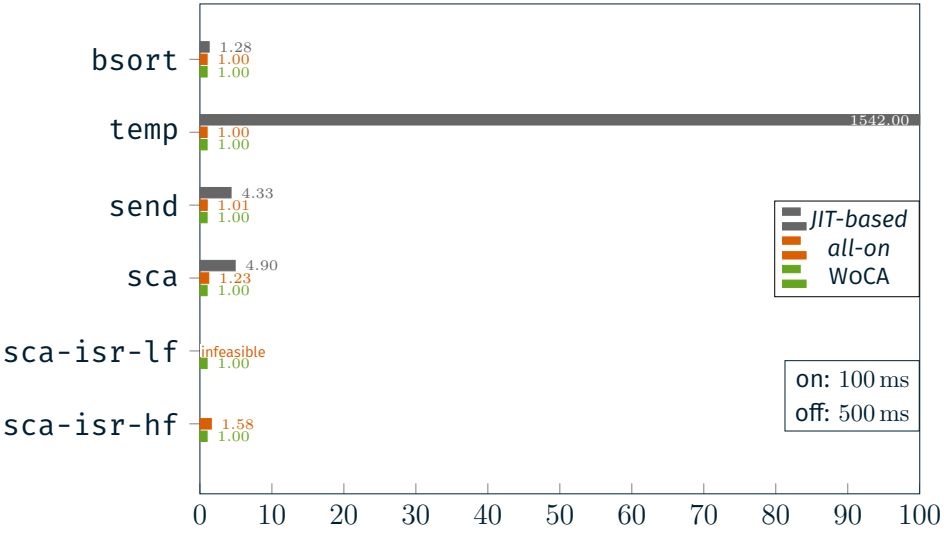
Results: Internal Device State



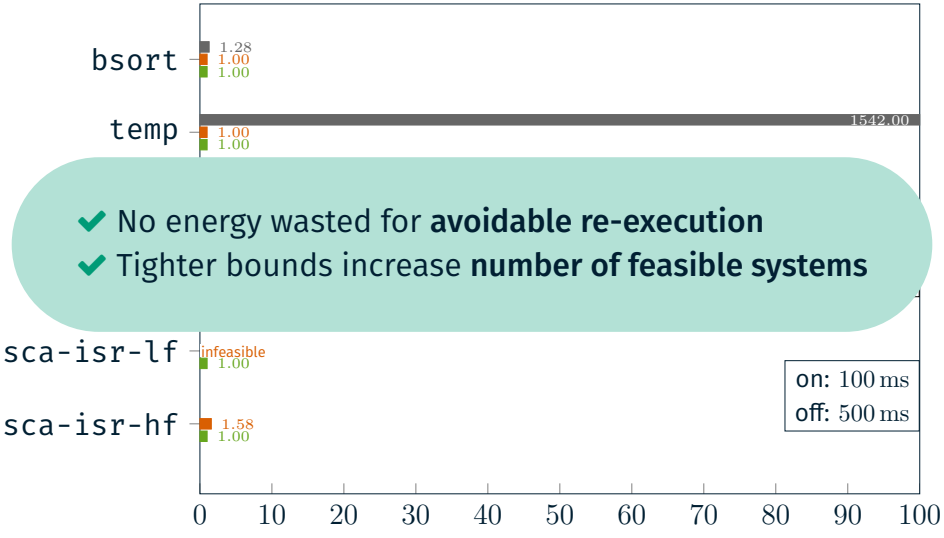
Results: Internal Device State



Results: Intermittent Execution



Results: Intermittent Execution



- ✔ No energy wasted for **avoidable re-execution**
- ✔ Tighter bounds increase **number of feasible systems**

Conclusion

Conclusion

- Problem of **device states** with **varying power demand**
- Problem of **asynchronous** device use in **different contexts**
- Problem of **re-execution** due to transactional operation semantics

Conclusion

- ✔ Problem of **device states** with **varying power demand**
  - Modeling of devices ⇝ Device Graph
- ⚠ Problem of **asynchronous** device use in **different contexts**
- ⚠ Problem of **re-execution** due to transactional operation semantics

Conclusion

- ✔ Problem of **device states** with **varying power demand**
  - Modeling of devices ⇝ Device Graph
- ✔ Problem of **asynchronous** device use in **different contexts**
  - Context-aware modeling of device usage ⇝ System Graph
- ⚠ Problem of **re-execution** due to transactional operation semantics

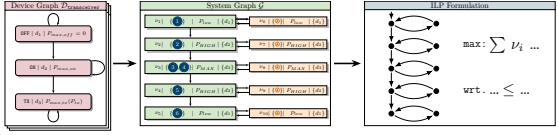
Conclusion

- ✔ Problem of **device states** with **varying power demand**
  - Modeling of devices $\rightsquigarrow$ Device Graph
- ✔ Problem of **asynchronous** device use in **different contexts**
  - Context-aware modeling of device usage $\rightsquigarrow$ System Graph
- ✔ Problem of **re-execution** due to transactional operation semantics
  - Transactional guarantees through analysis bounds

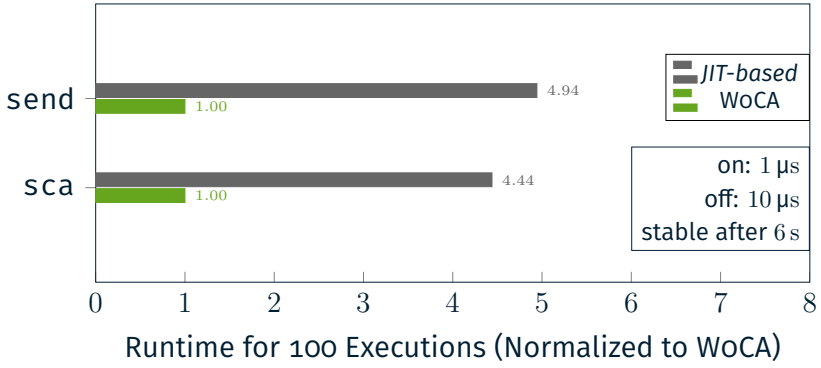
Conclusion

- Problem of **device states** with **varying power demand**
  - Modeling of devices ⇝ Device Graph
- Problem of **asynchronous** device use in **different contexts**
  - Context-aware modeling of device usage ⇝ System Graph
- Problem of **re-execution** due to transactional operation semantics
  - Transactional guarantees through analysis bounds

WoCA's hard- & software:
https://gitos.rrze.fau.de/woca



Results: Starvation Freedom



Experiment Observations

- Power supply constantly on after $6\,\mathrm{s}$
- *JIT-based* only makes significant progress after supply is stable
- WoCA **avoids starvation**