Towards Power-aware Clock Configuration for Embedded Operating Systems

Michel Rottleuthner
michel.rottleuthner@haw-hamburg.de

Hamburg University of Applied Sciences
https://inet.haw-hamburg.de/

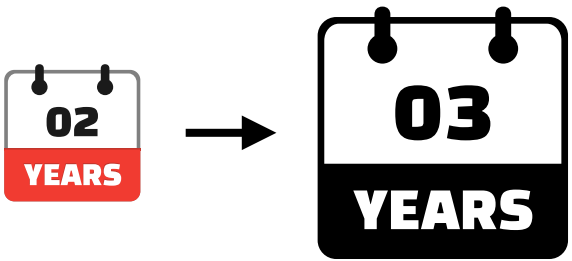




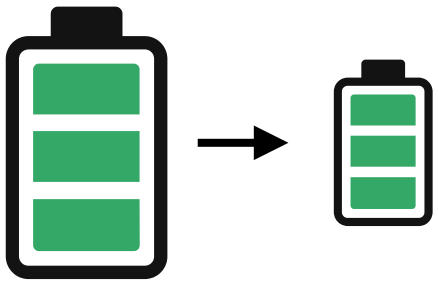
Outline

- Fundamentals
- What is needed for Dynamic Clock Configuration?
  - Abstraction and Runtime Control
- Obtainable Energy Savings for IoT Networking
- Clock Tree Power Modeling
- Conclusion and Outlook

Motivation: Maximize Energy-efficiency



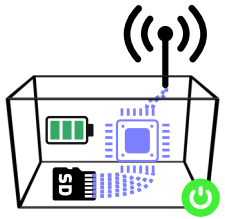
Increase lifetime



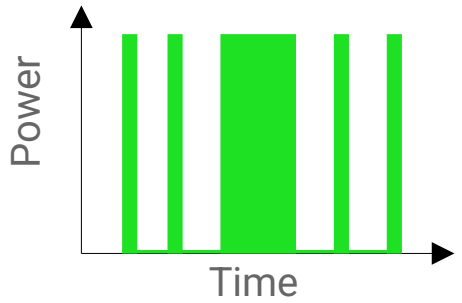
reduce battery size

How to do this without significant effort from application developers?

Recap: Duty-Cycling vs. DVFS



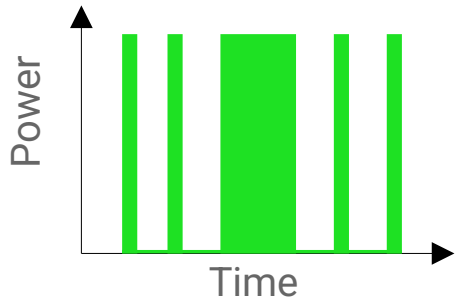
- Duty Cycling
  - Put system to sleep on idle
  - Wakeup via timers or interrupts
  - No processing during sleep
  - Complexity: low



Recap: Duty-Cycling vs. DVFS



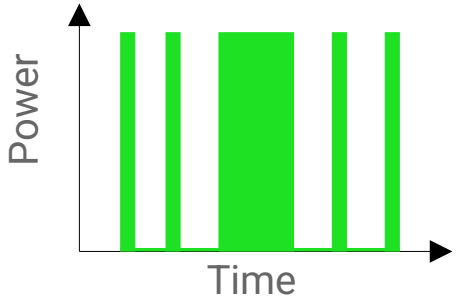
- Duty Cycling
  - Put system to sleep on idle
  - Wakeup via timers or interrupts
  - No processing during sleep
  - Complexity: low

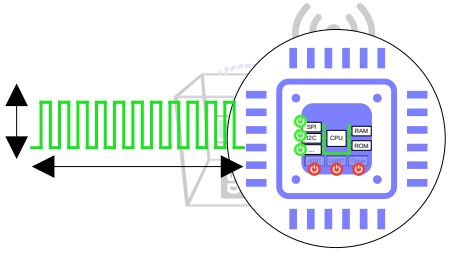


Recap: Duty-Cycling vs. DVFS

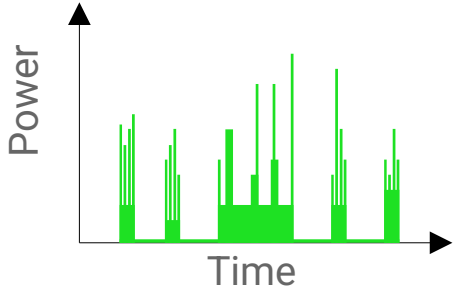


- Duty Cycling
  - Put system to sleep on idle
  - Wakeup via timers or interrupts
  - No processing during sleep
  - Complexity: low

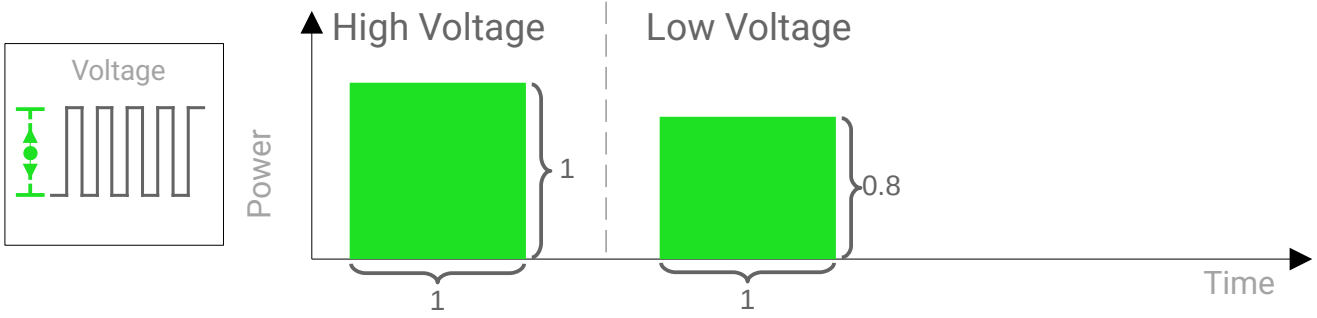




- Dynamic Voltage and Frequency Scaling
  - Fine grained performance control
  - Low-power processing
  - Complexity: very high



Dynamic Voltage and Frequency Scaling



Energy

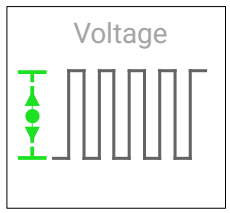
$$W = P \cdot t$$

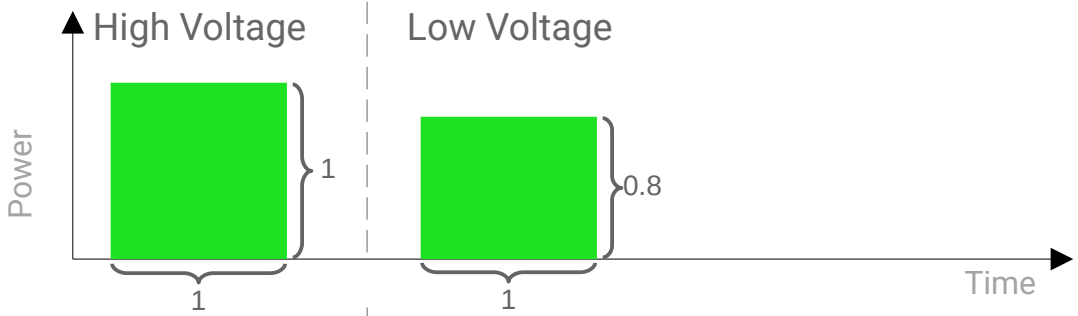
1 → 0.8

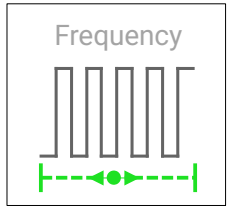
„Objective Function!“

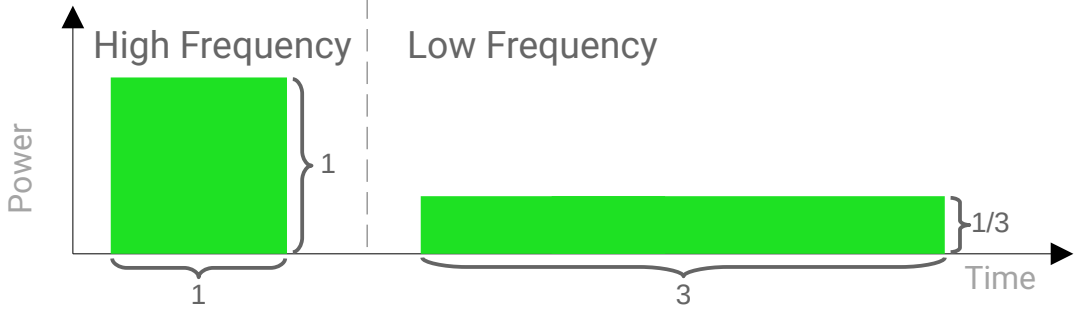
Dynamic Voltage and Frequency Scaling

Task scales with frequency









Energy

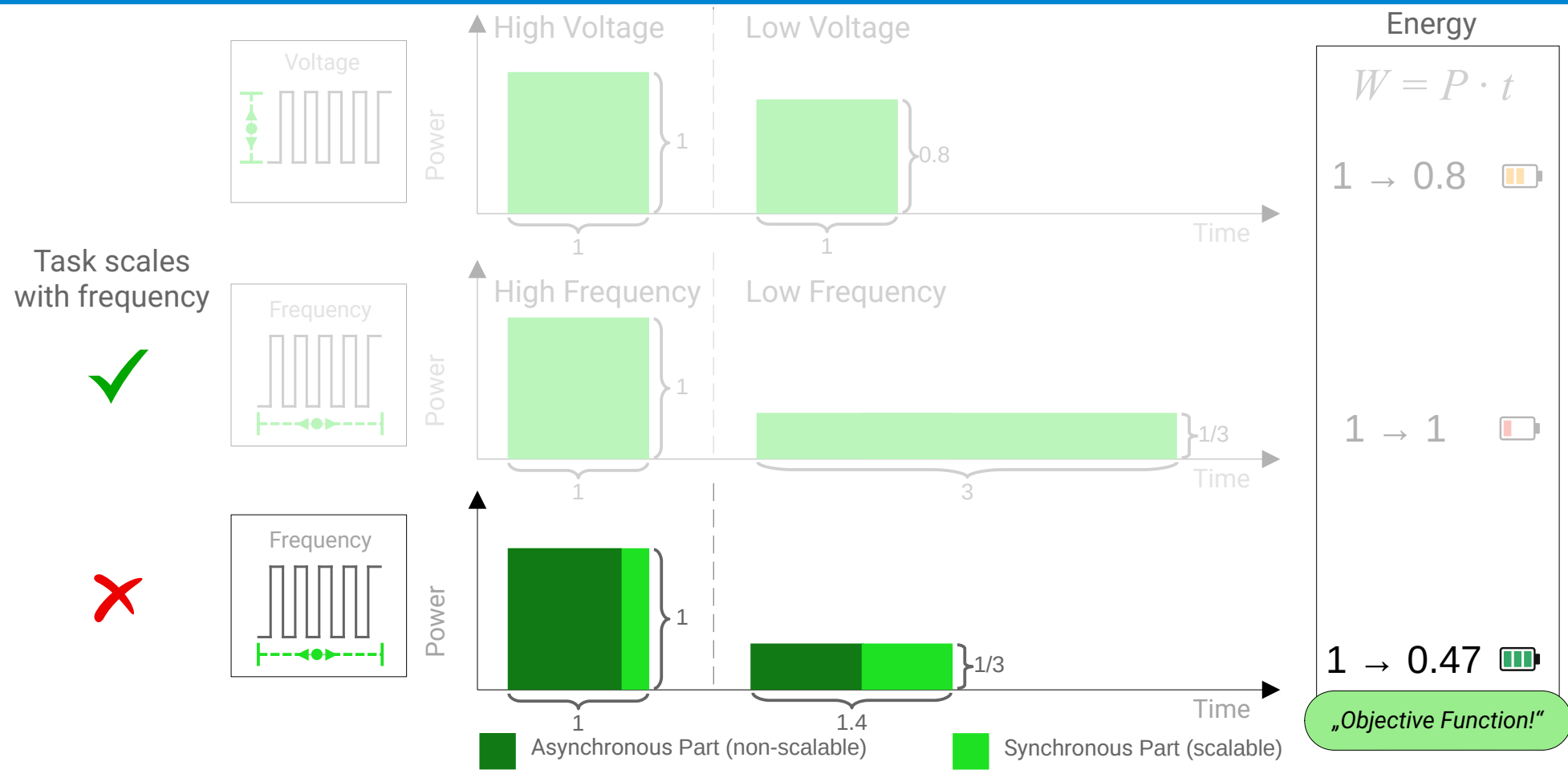
$$W = P \cdot t$$

1 → 0.8

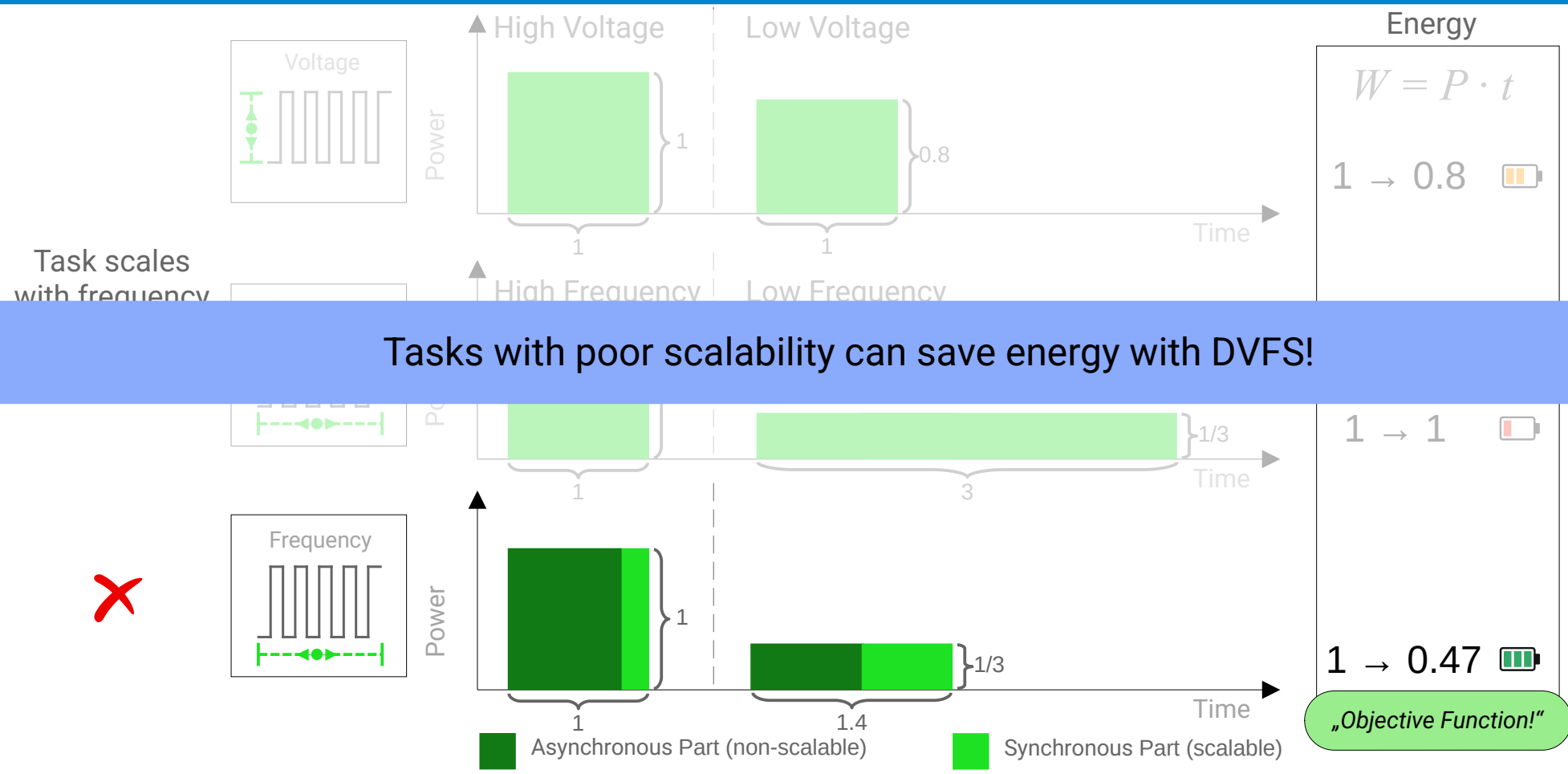
1 → 1

„Objective Function!“

Dynamic Voltage and Frequency Scaling



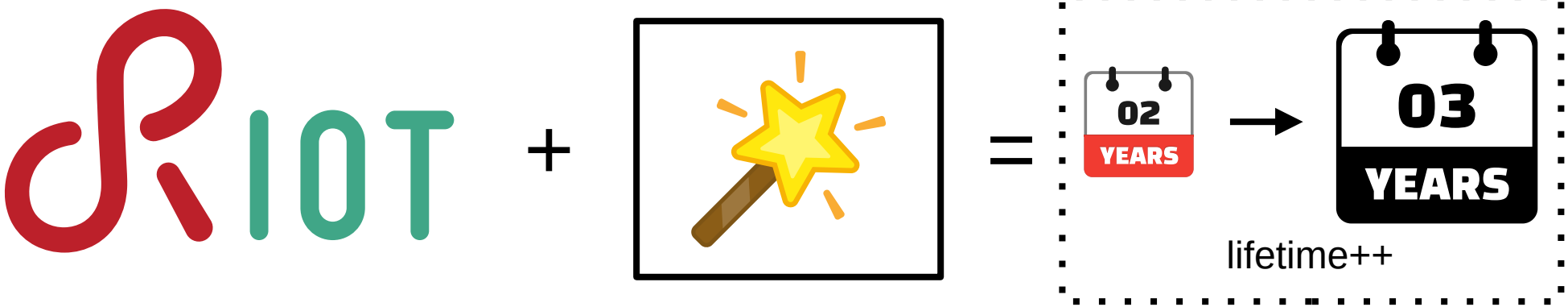
Dynamic Voltage and Frequency Scaling



Tasks with poor scalability can save energy with DVFS!

The Vision

```
USEMODULE += auto_dvfs
```

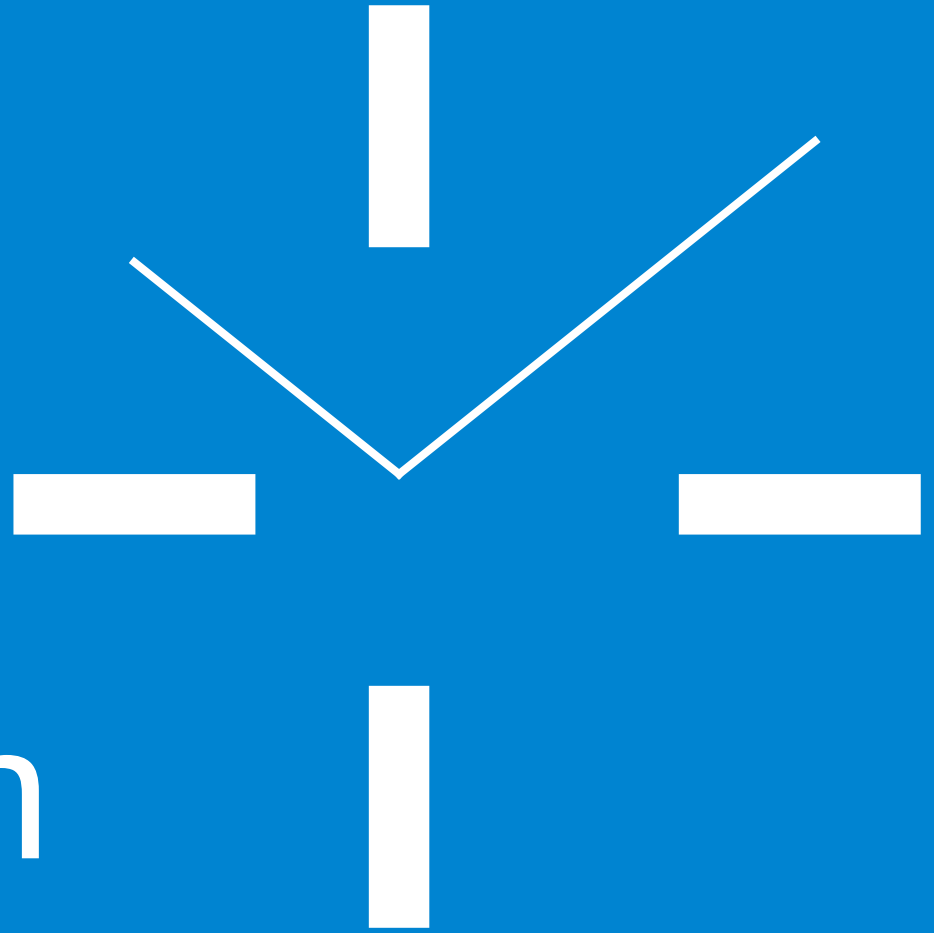


What do we need for `auto_dvfs` ()?

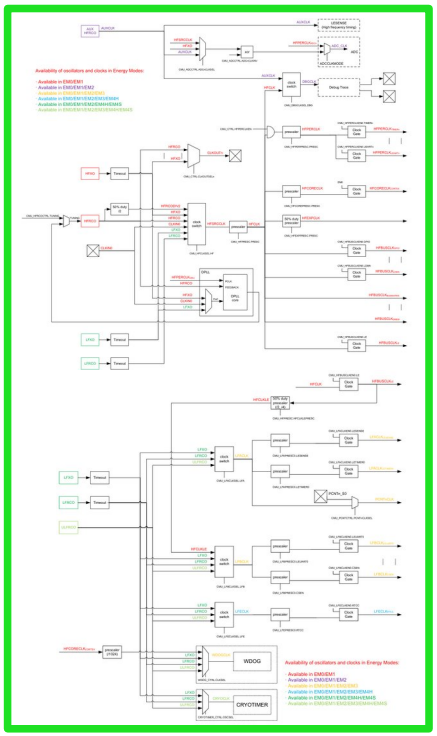
- Generic Clock Configuration Abstraction
- DVFS Control Method(s)
  - Offline, Online?
- Experiments
- Thorough understanding and generalization …
  - … of the hardware properties
  - … of the software scalability

} and how they affect each other

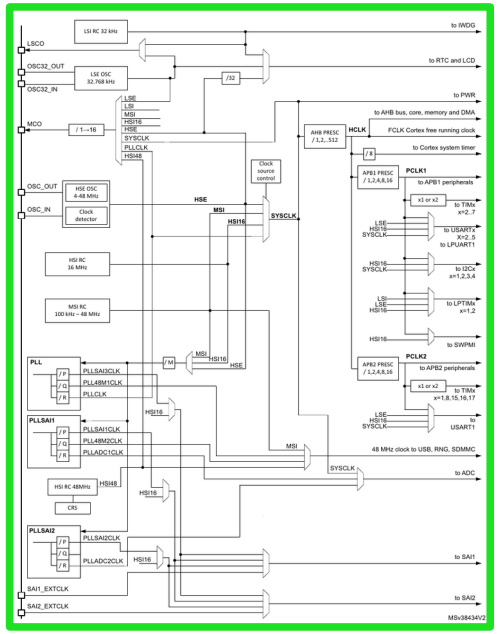
The Clock-Configuration Abstraction



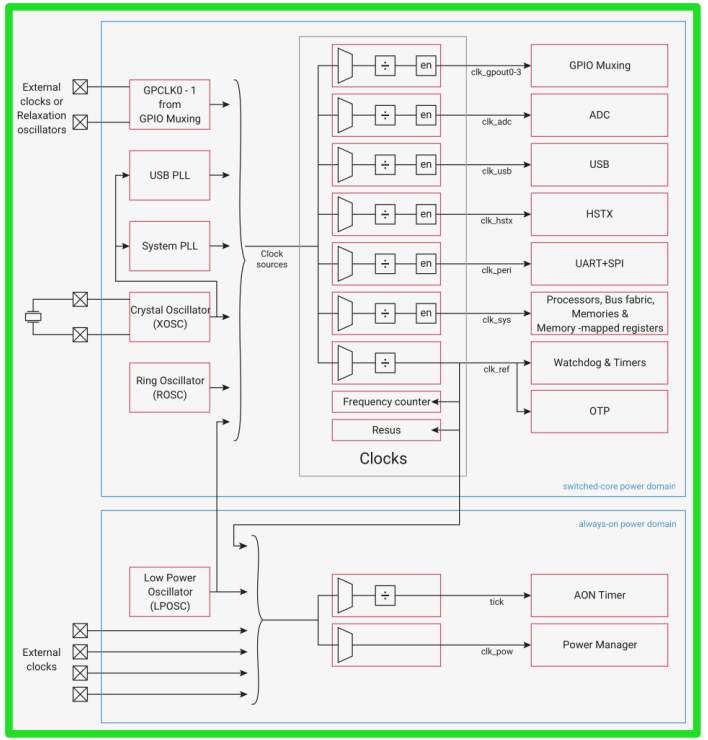
Clock Trees



EFM32PG12B



STM32L476RG

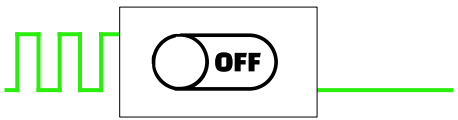


RP2350

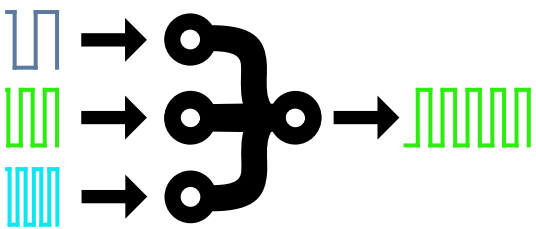
Shared Functionality of Clock Trees

"Generic Clock Nodes"

Gate



Mux

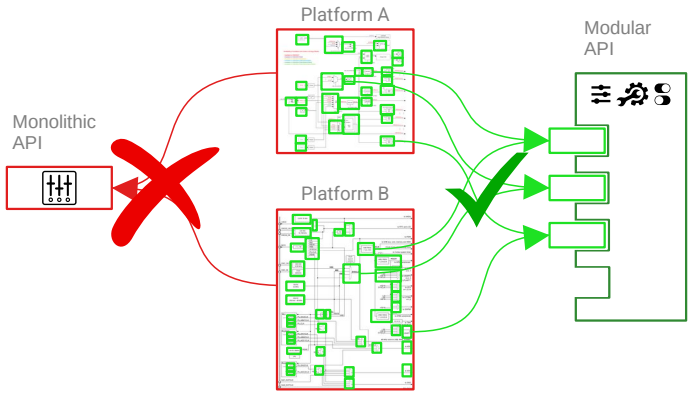


Scaler

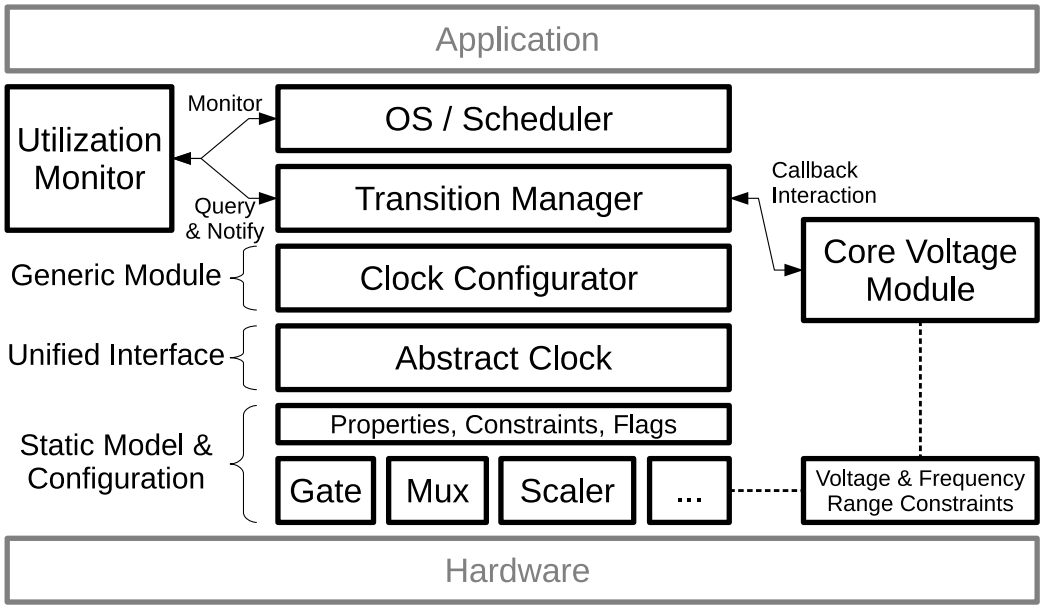


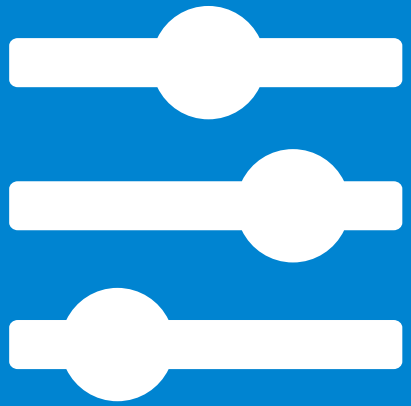
ScaleClock

Clock Tree Abstraction



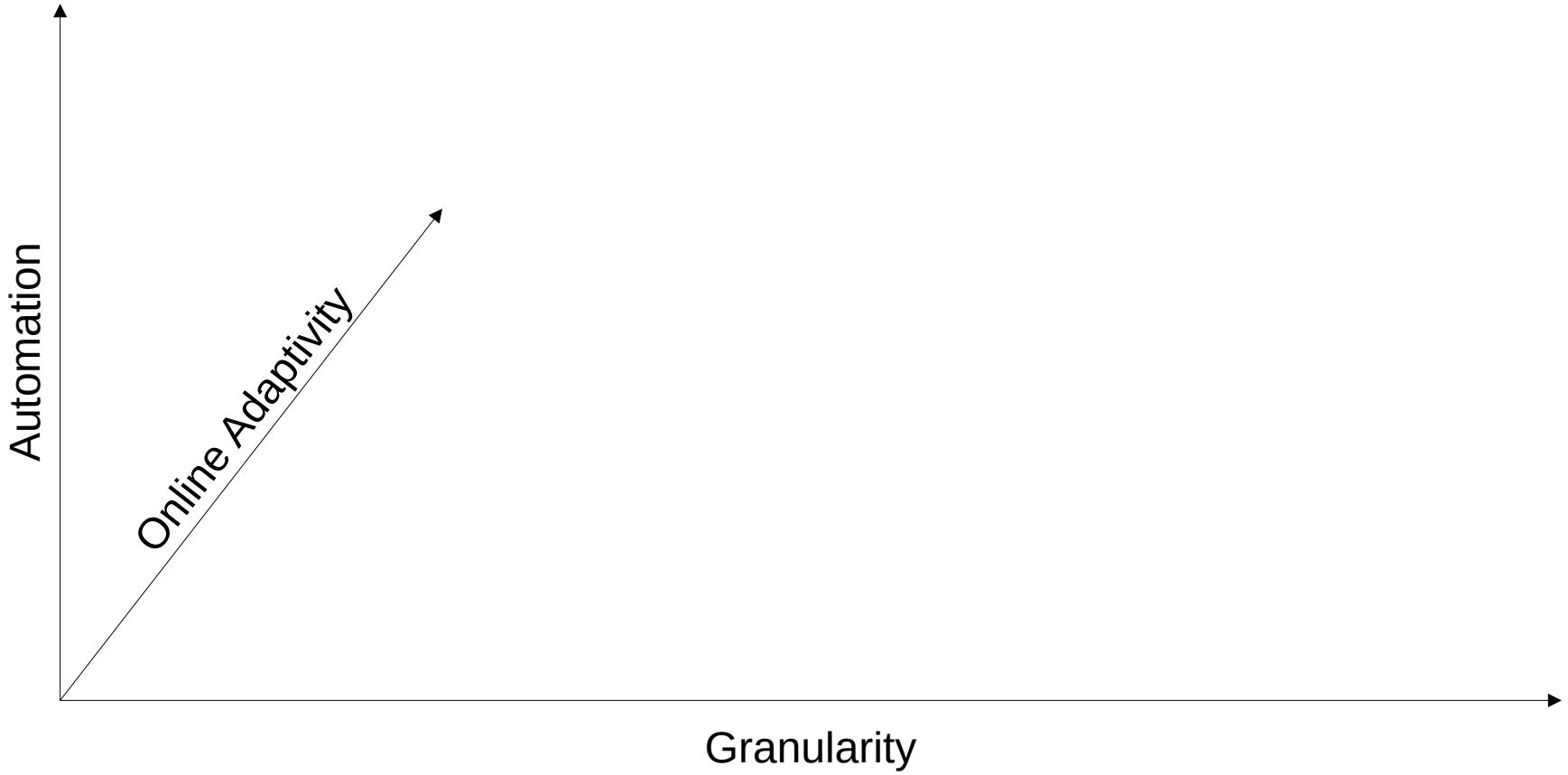
Architecture & OS-Integration



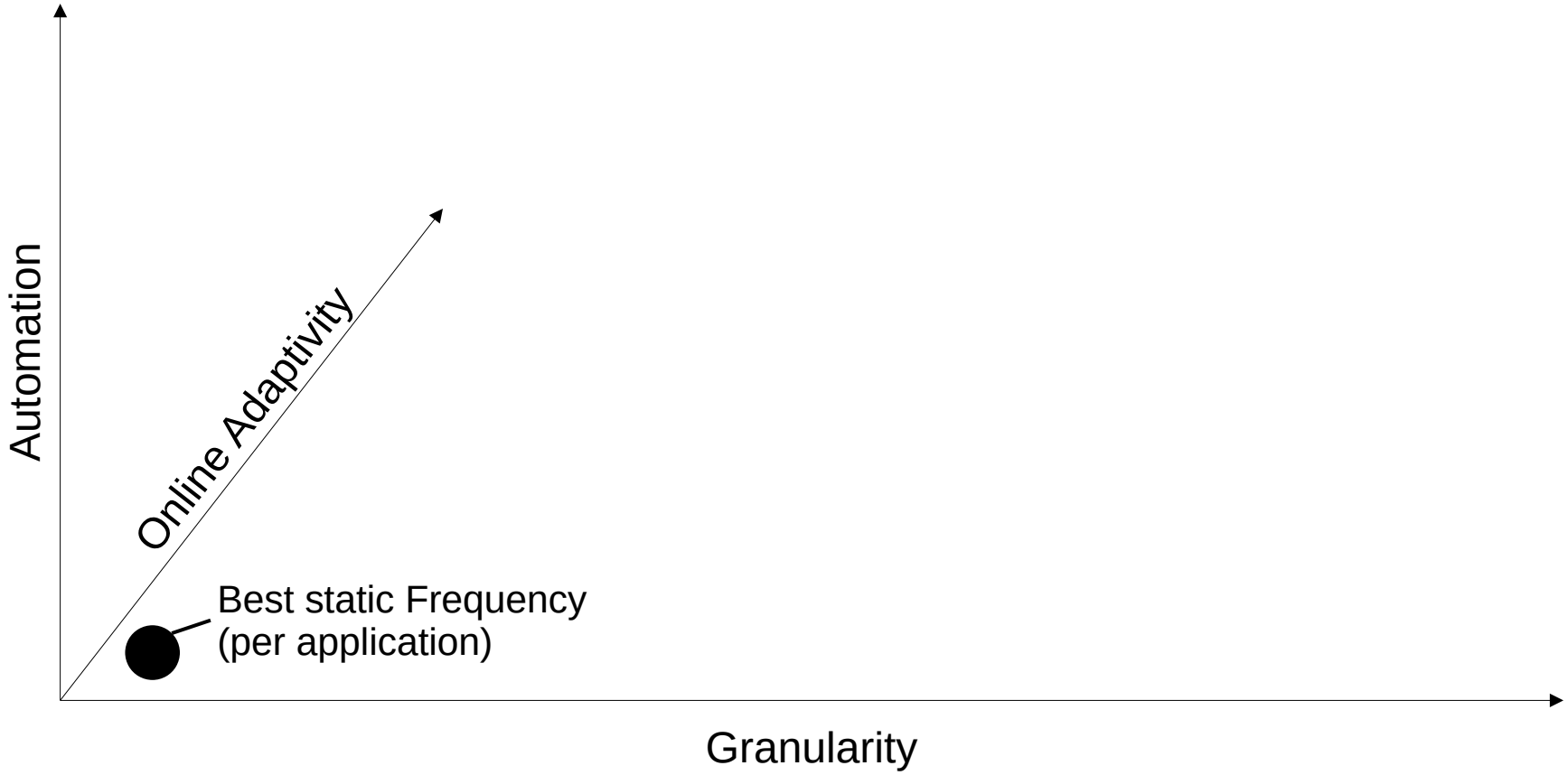


DVFS Control Methods

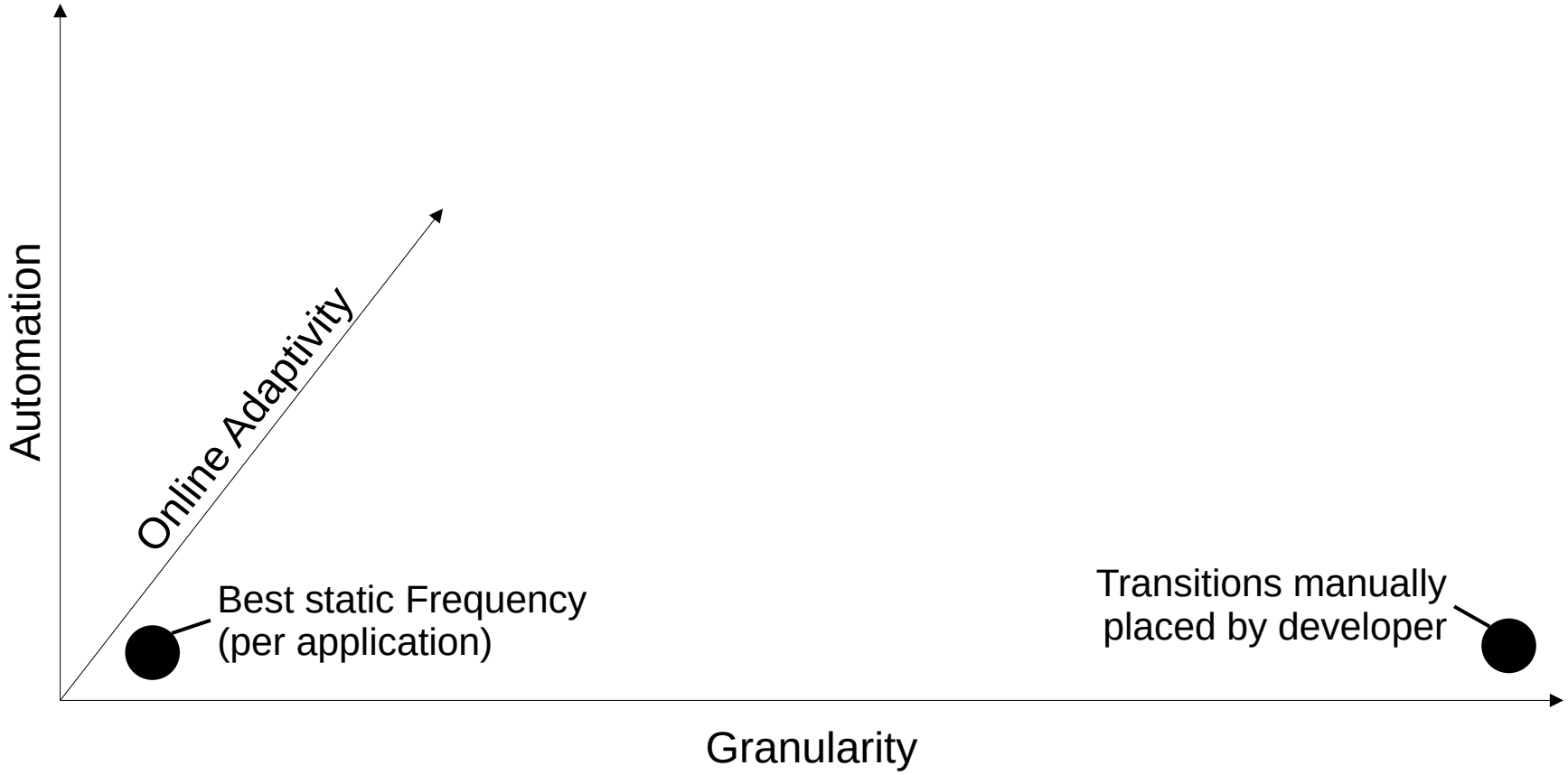
DVFS Control



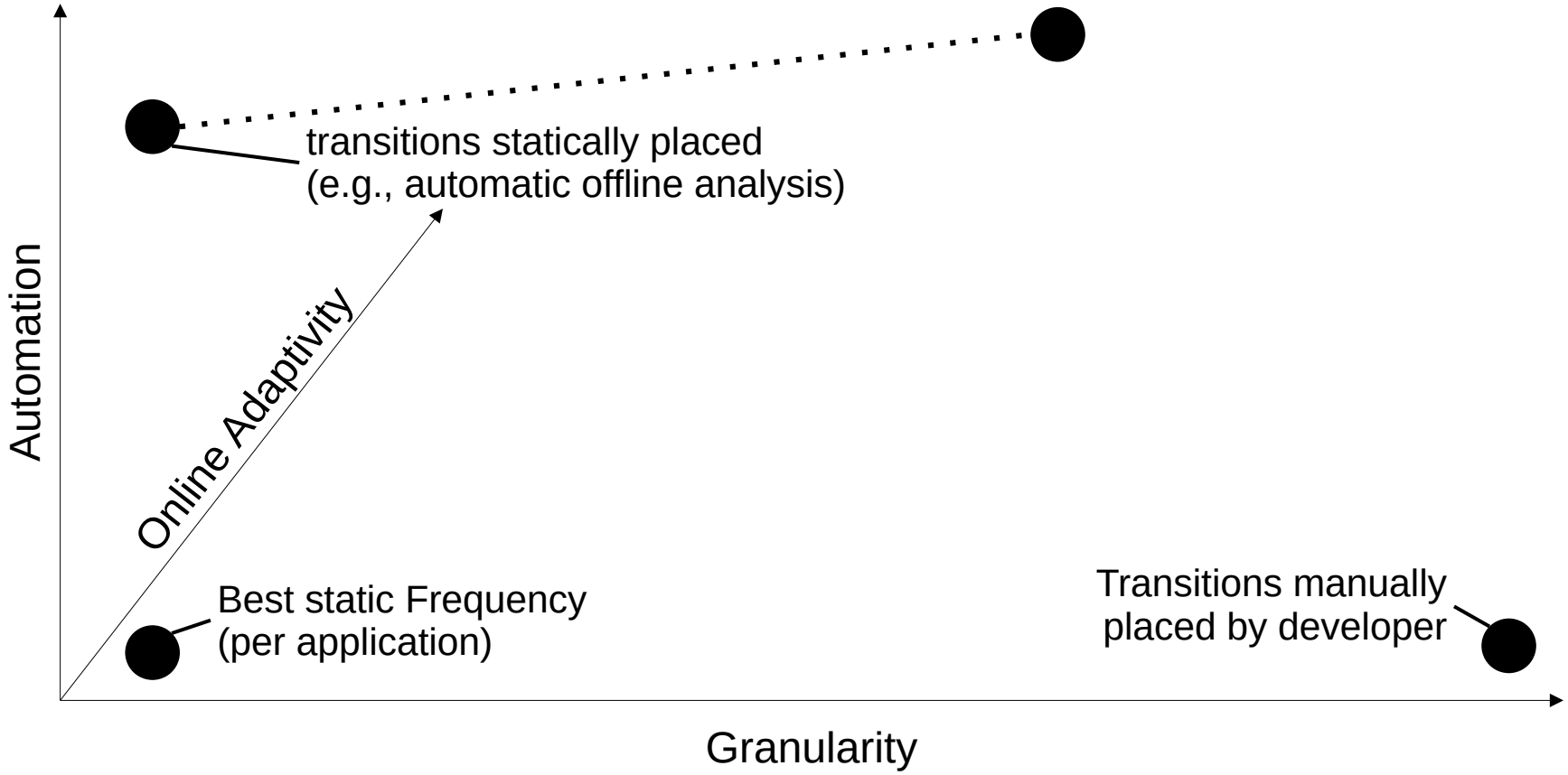
DVFS Control



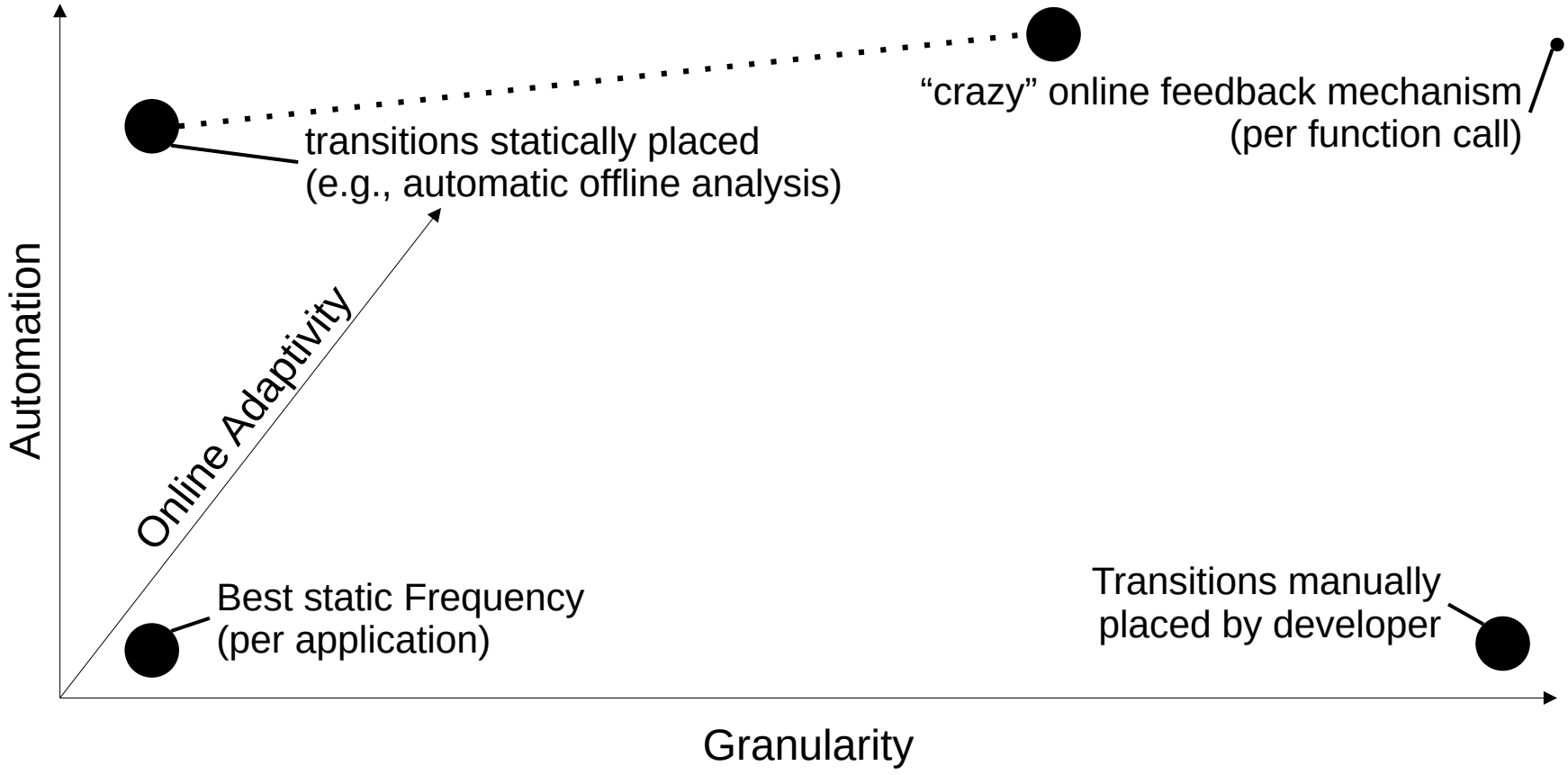
DVFS Control



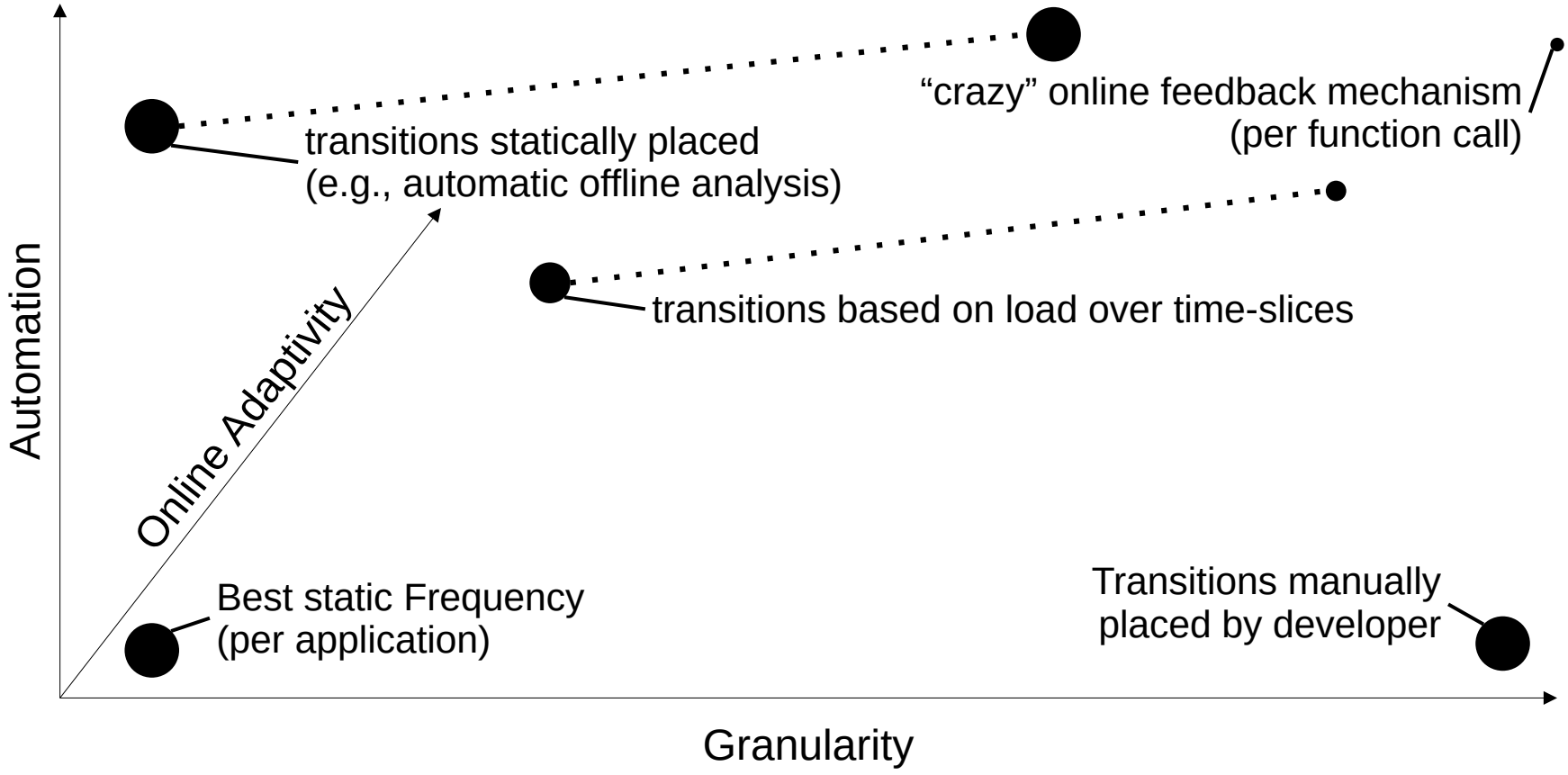
DVFS Control



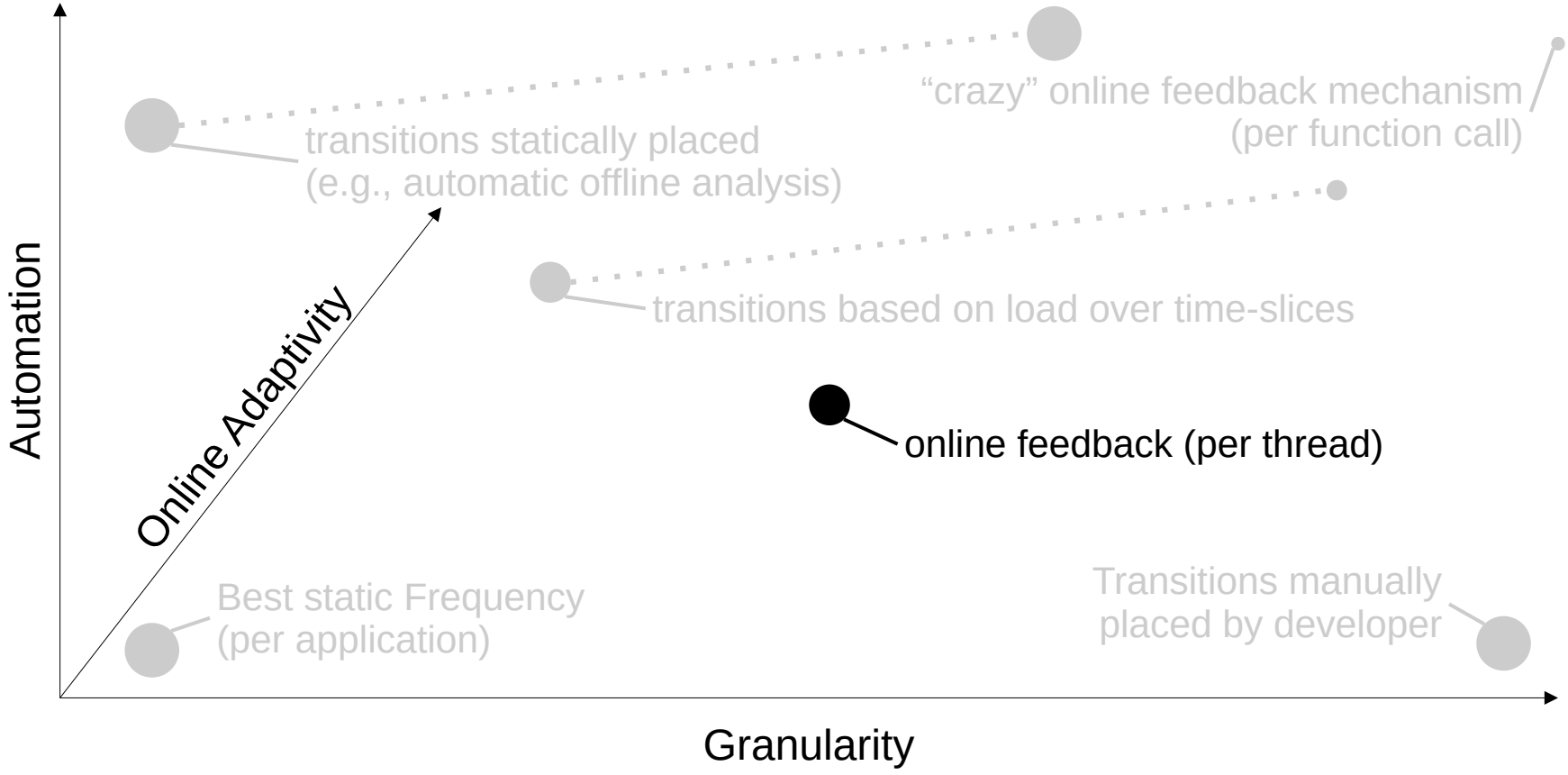
DVFS Control



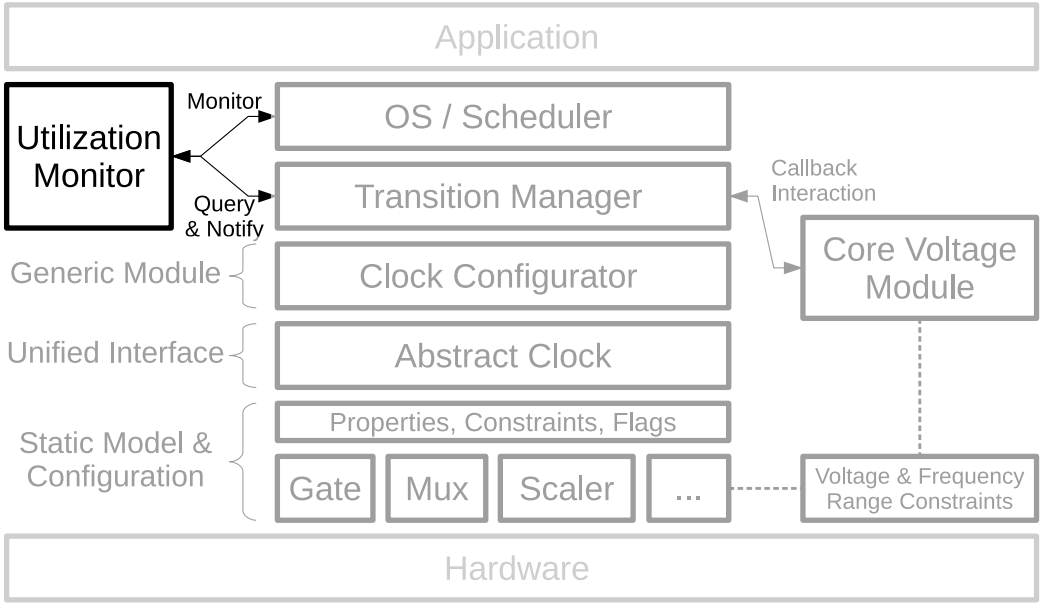
DVFS Control



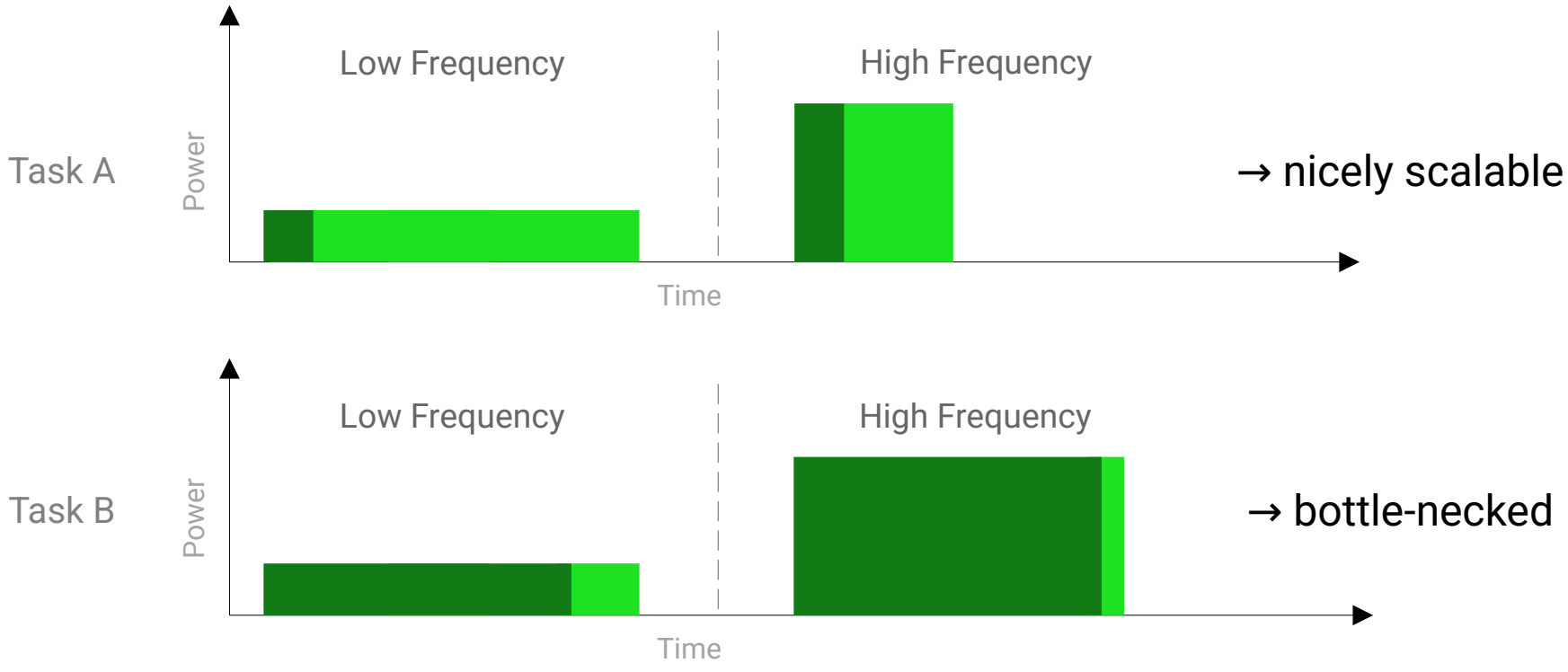
DVFS Control



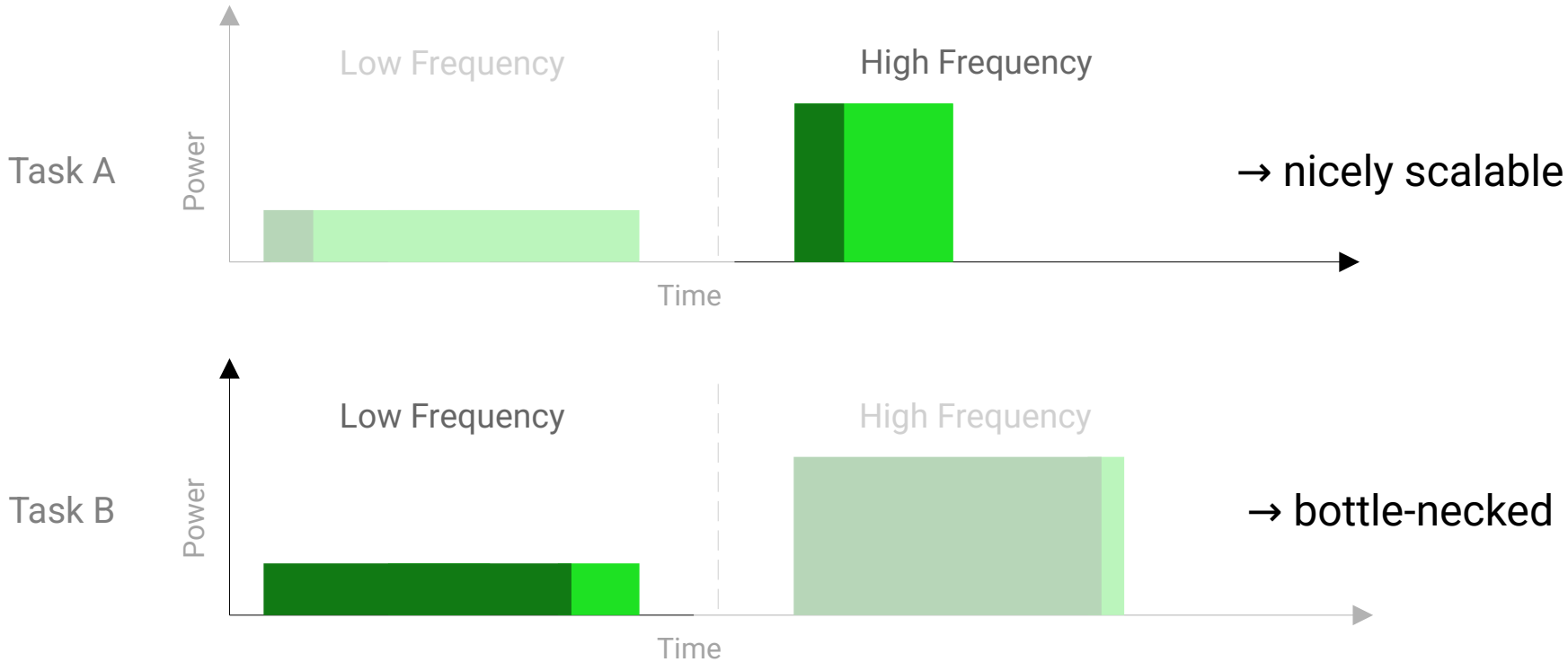
DVFS Control Per Thread



Remember: Task Scalability

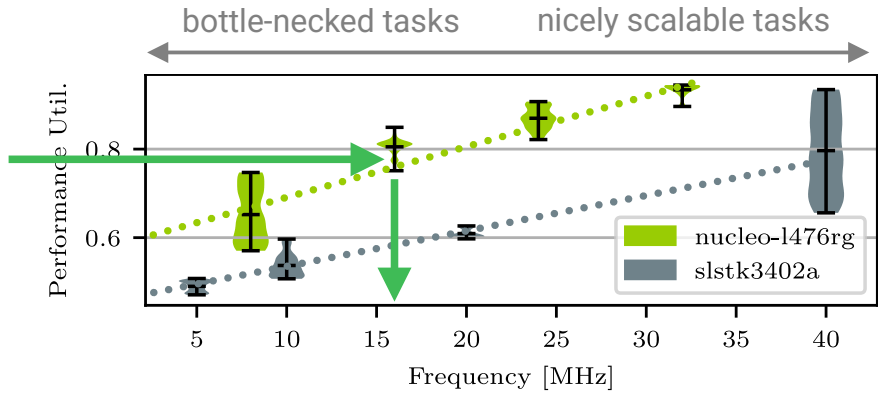


Remember: Task Scalability

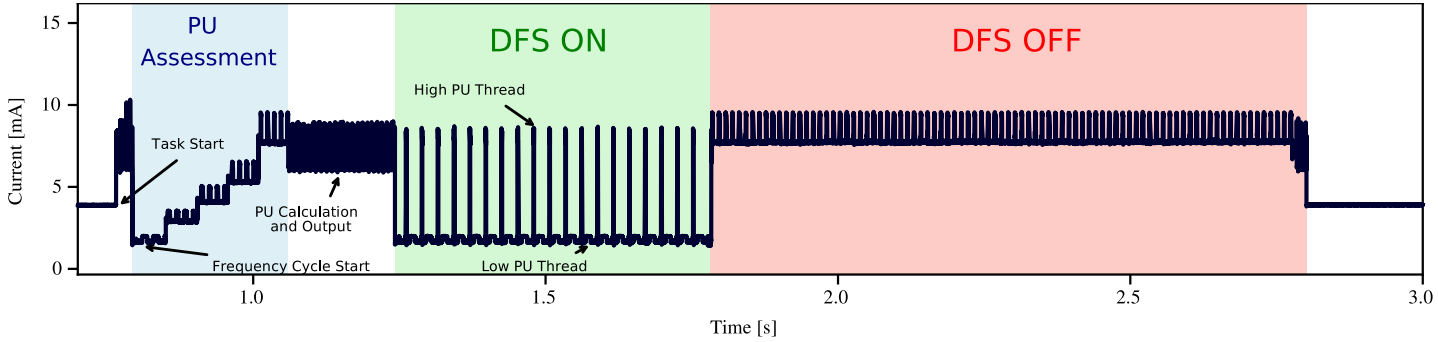


Performance Utilization Metric

$$PU = \frac{t_{busy}(F_1)}{t_{busy}(F_2)} \cdot \frac{F_1}{F_2}$$

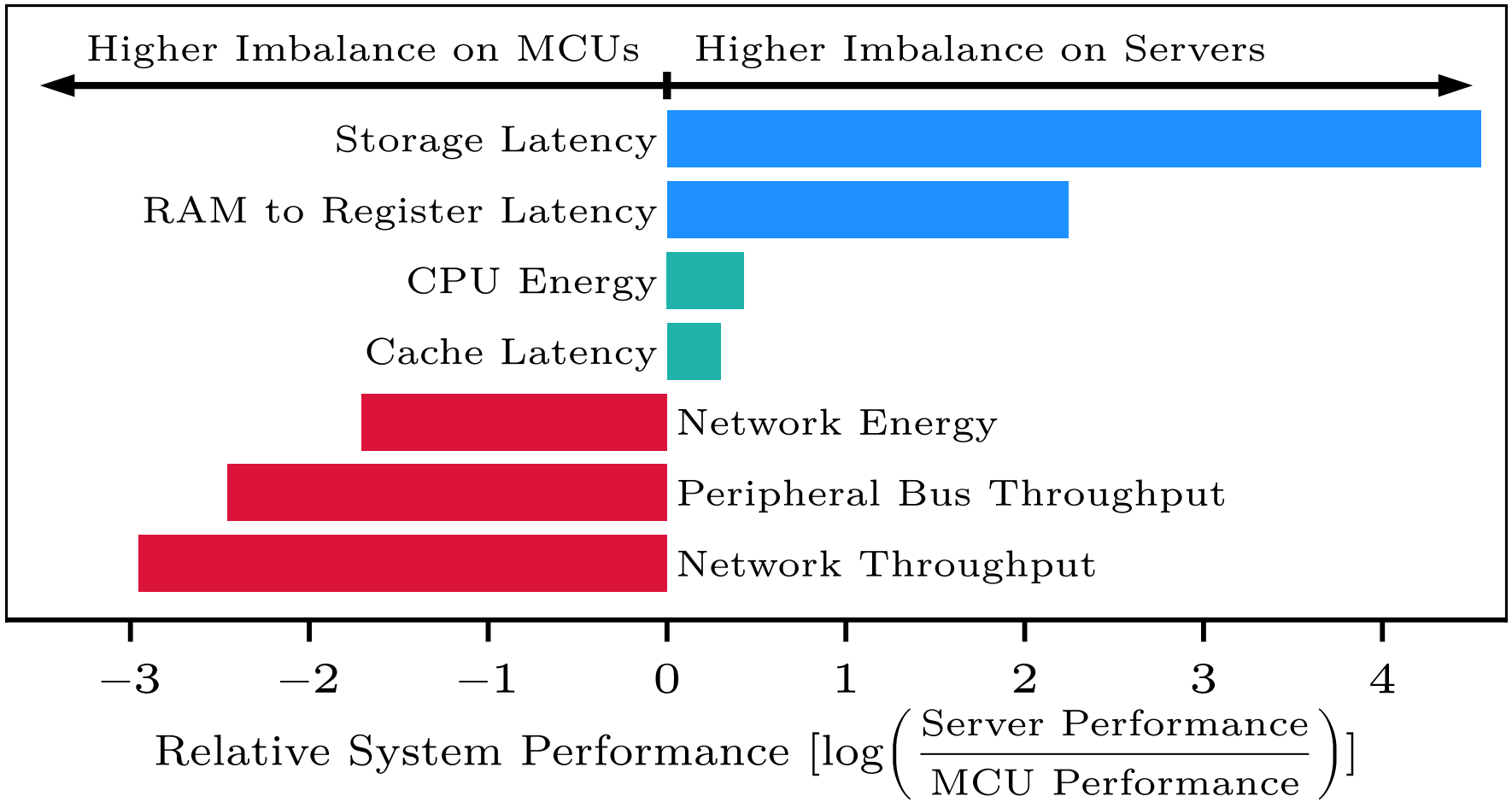


Apply dynamically: two tasks, one with low PU value (acquisition) and one with high PU value (processing)

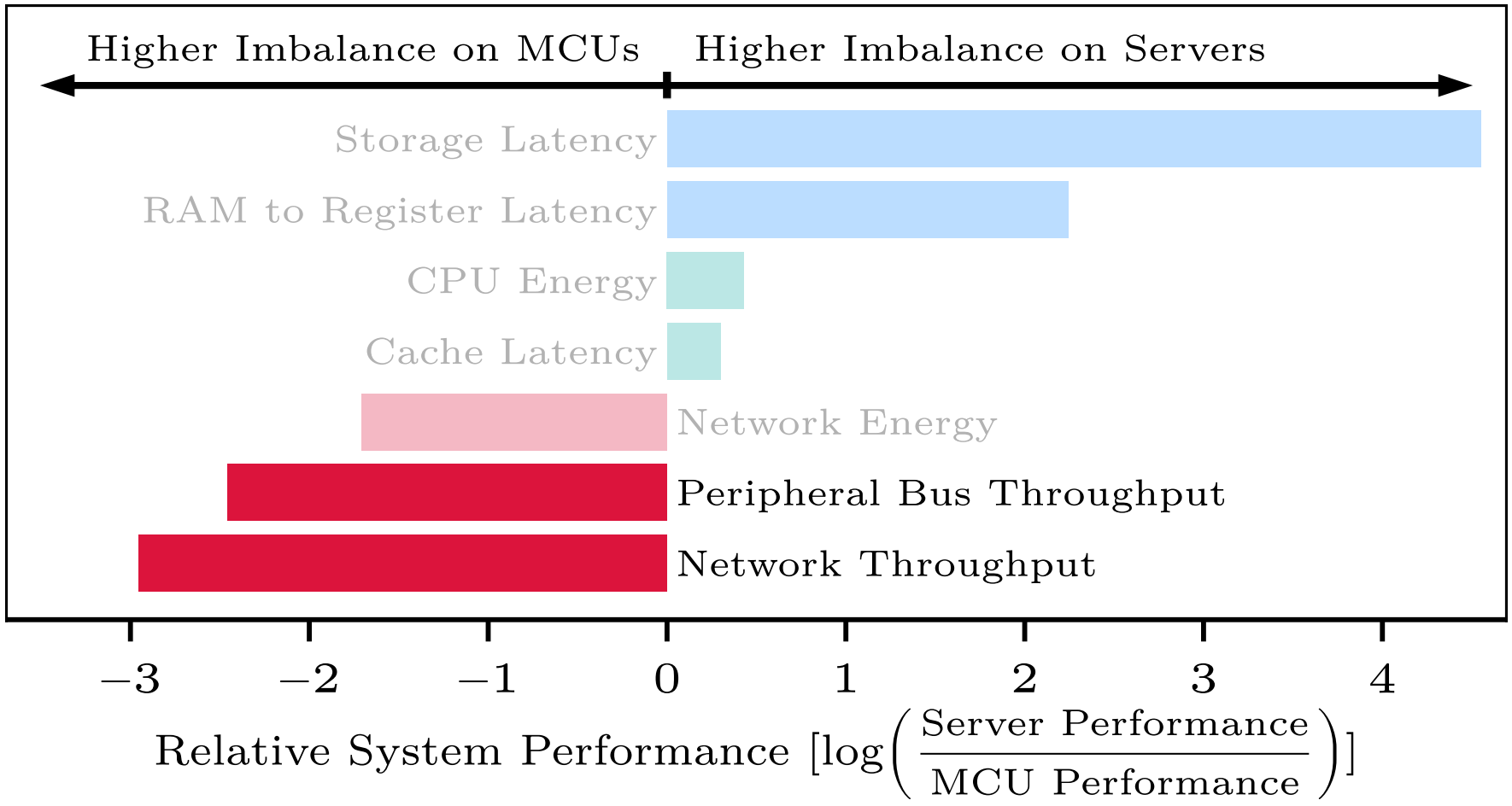


DVFS for IoT Applications

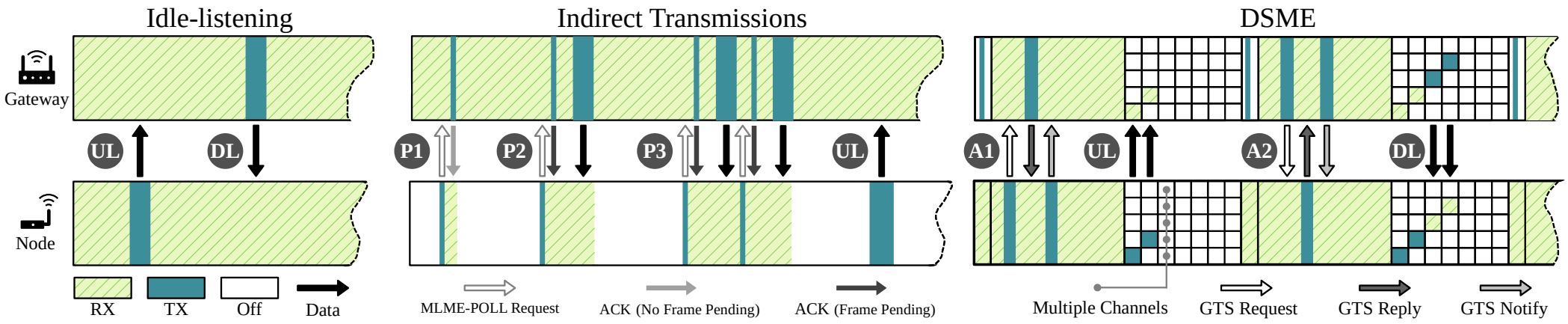
Subsystem Imbalance Suggests Networking



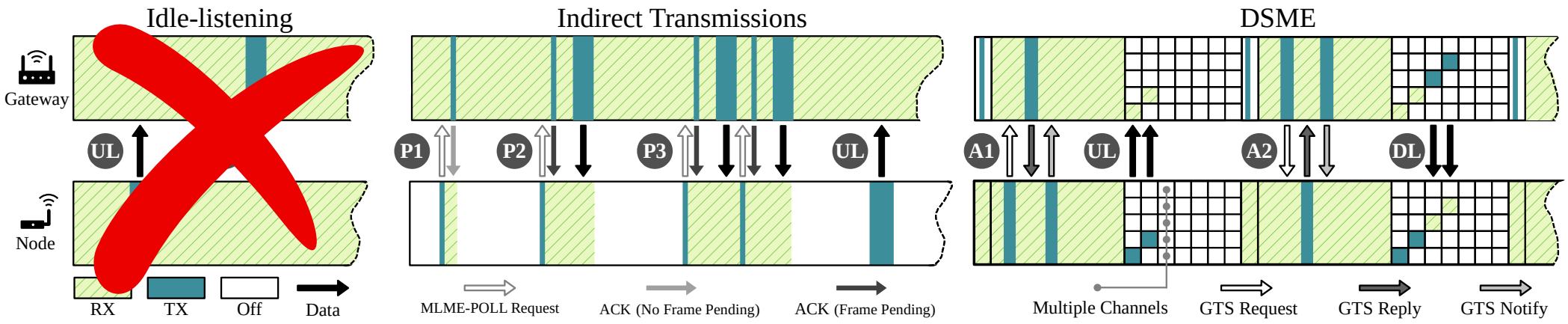
Subsystem Imbalance Suggests Networking



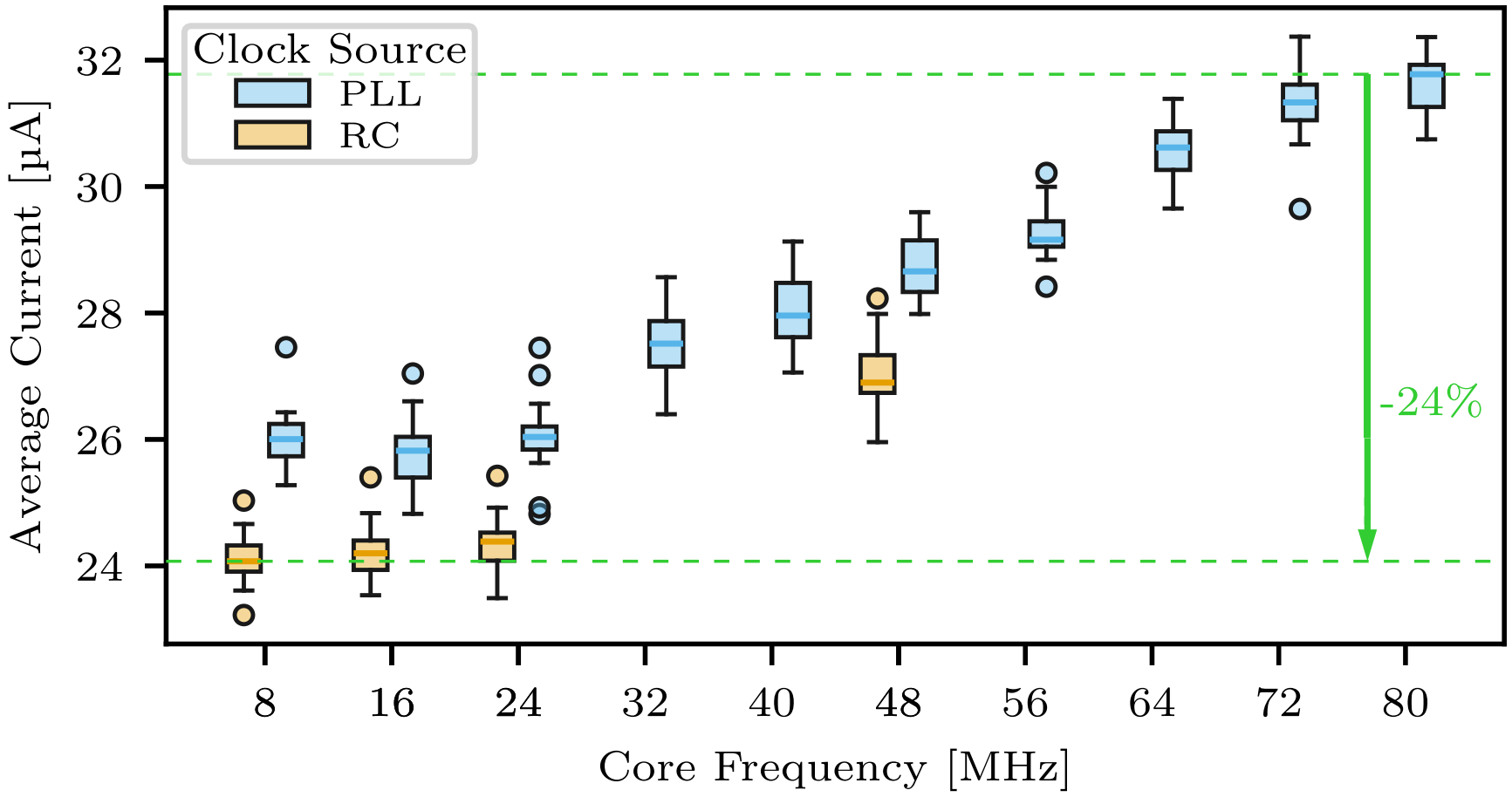
Experiments



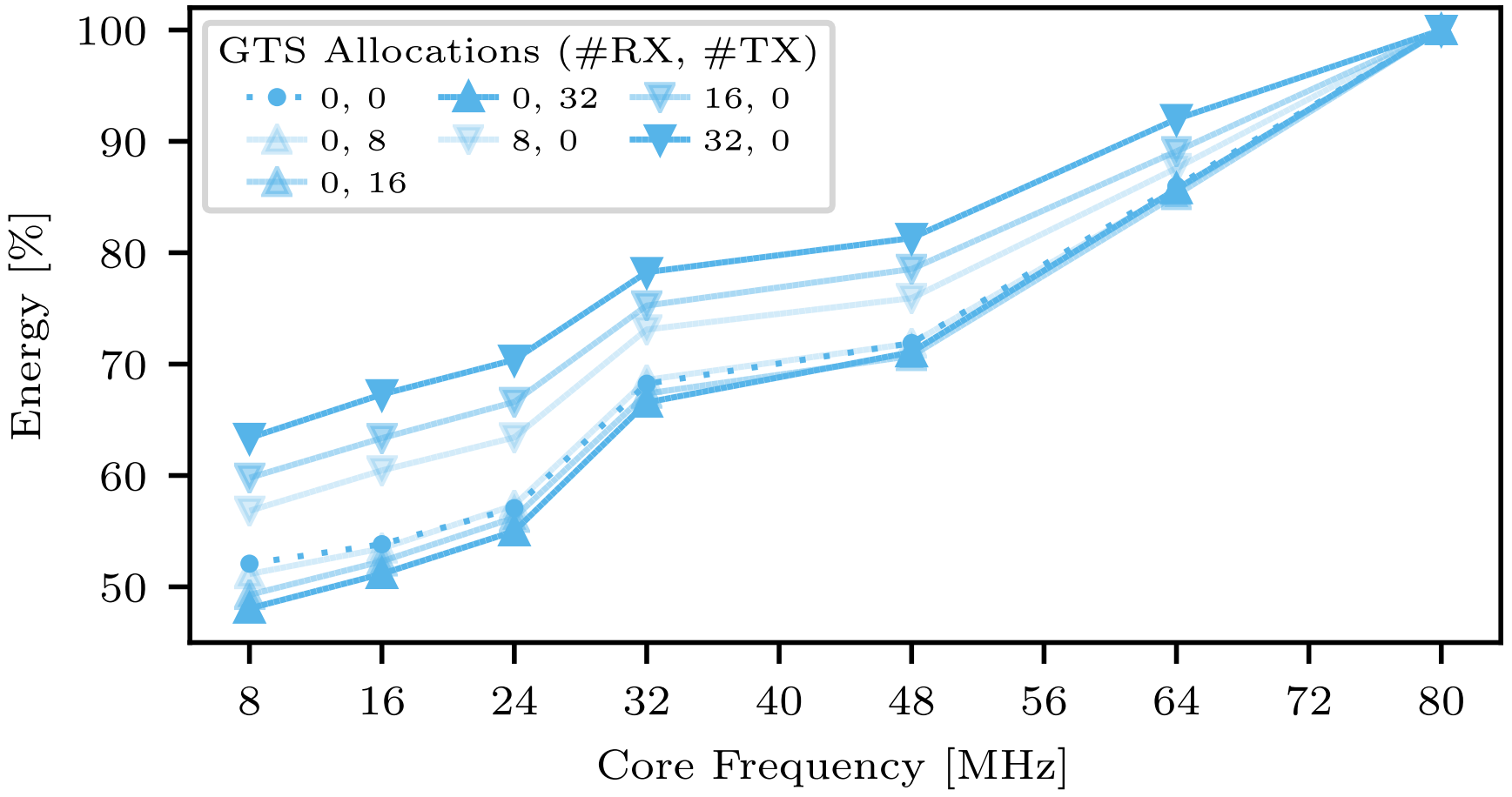
Experiments



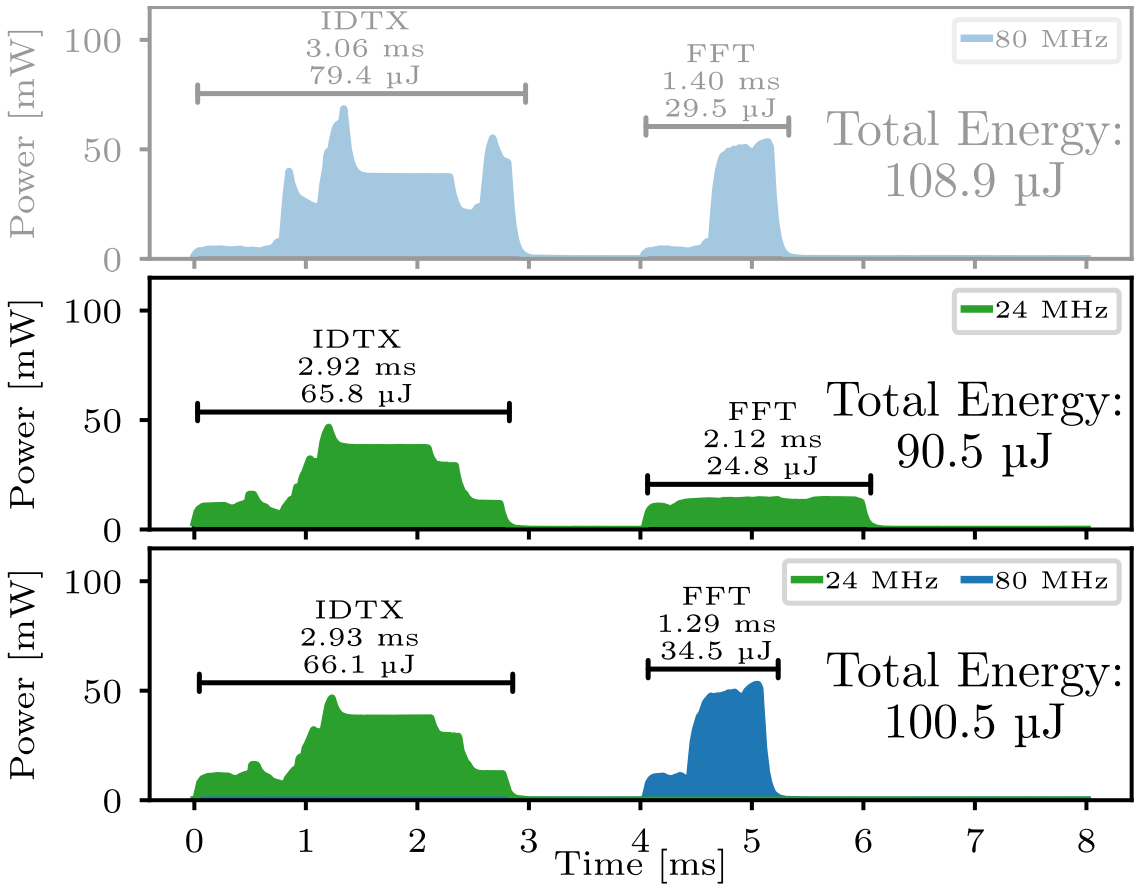
Indirect Transmissions (duty-cycling + DVFS)



DVFS for DSME



Multimodal Tasks (Indirect TX + FFT)



Static Default Frequency

Minimize Energy

Minimize Energy-Delay-Product

What do we need for `auto_dvfs` ()?

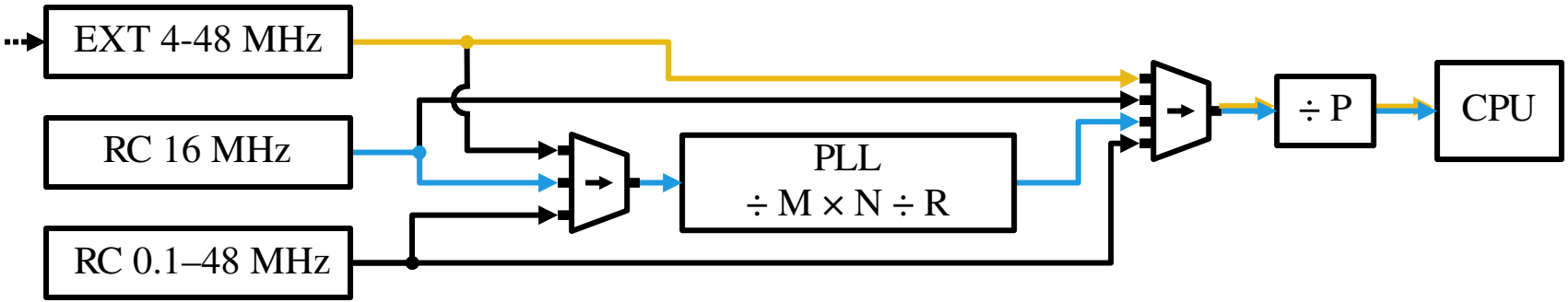
- Generic Clock Configuration Abstraction ✓
- DVFS Control Method(s) ✓
  - Offline, Online?
- Experiments ✓
- Thorough understanding and generalization …
  - … of the hardware properties
  - … of the software scalability

} and how they affect each other

Clock-Tree Power Modeling

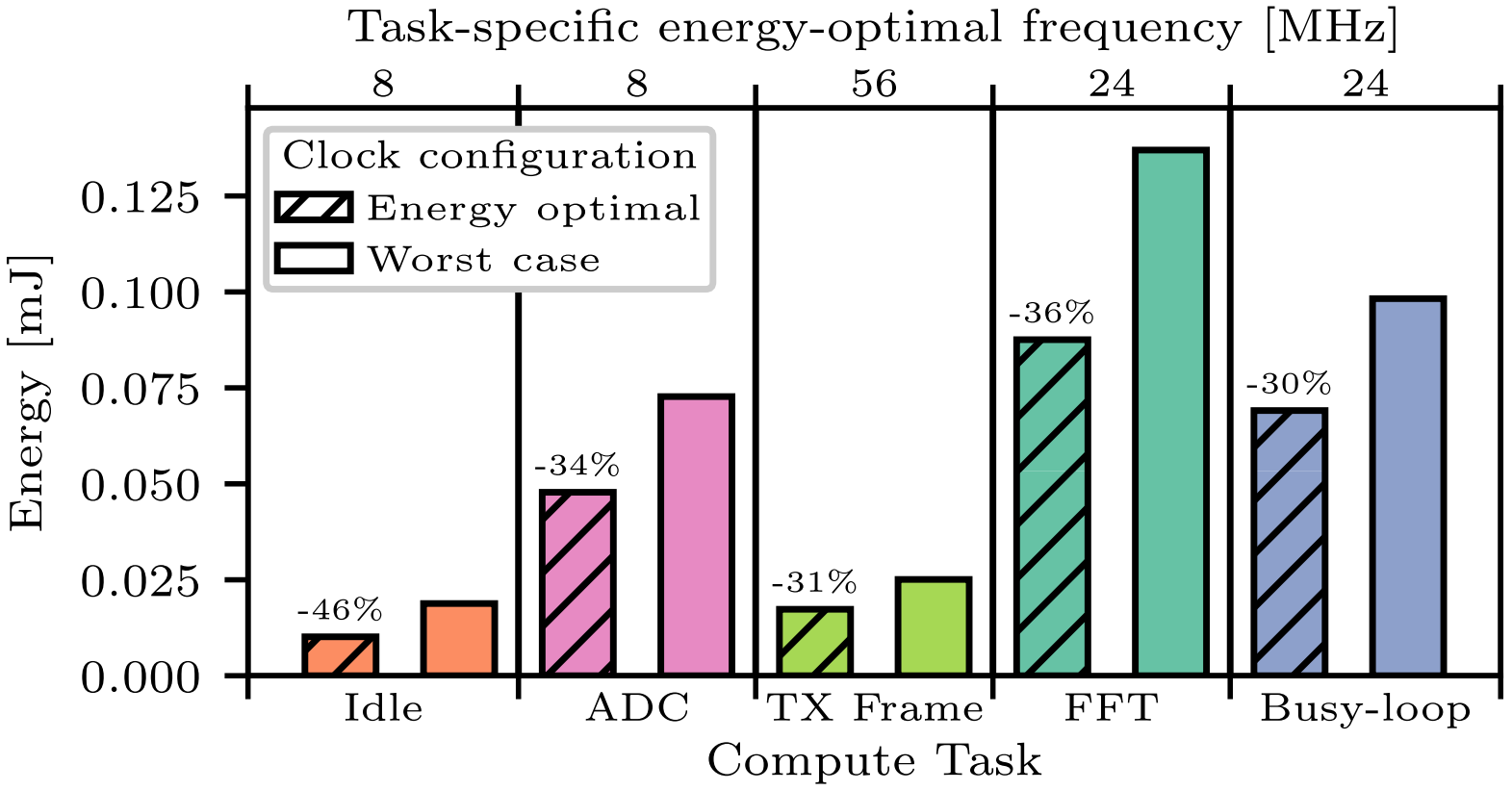


Same Frequency → Same Power ?

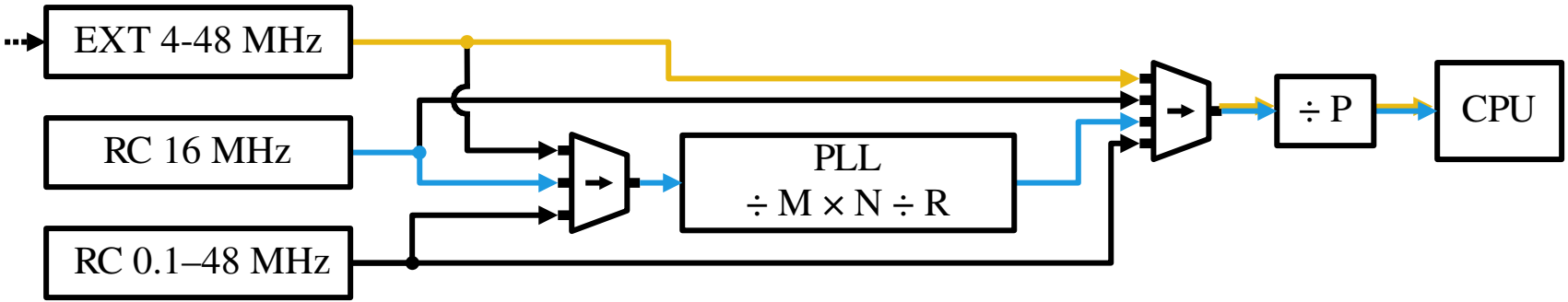


- 3 different Oscillators
- 6 different Topologies for clocking the CPU
- Up to thousands of configurations for generating the same frequency

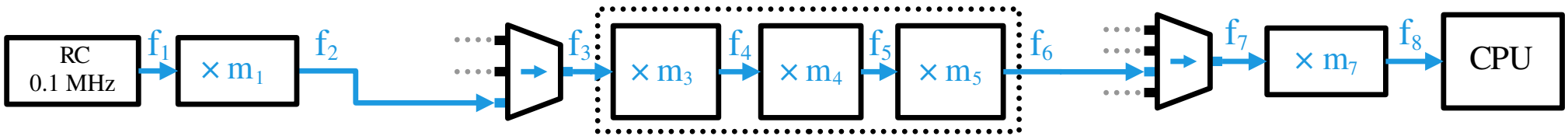
Topology and Factor Selection Matters!



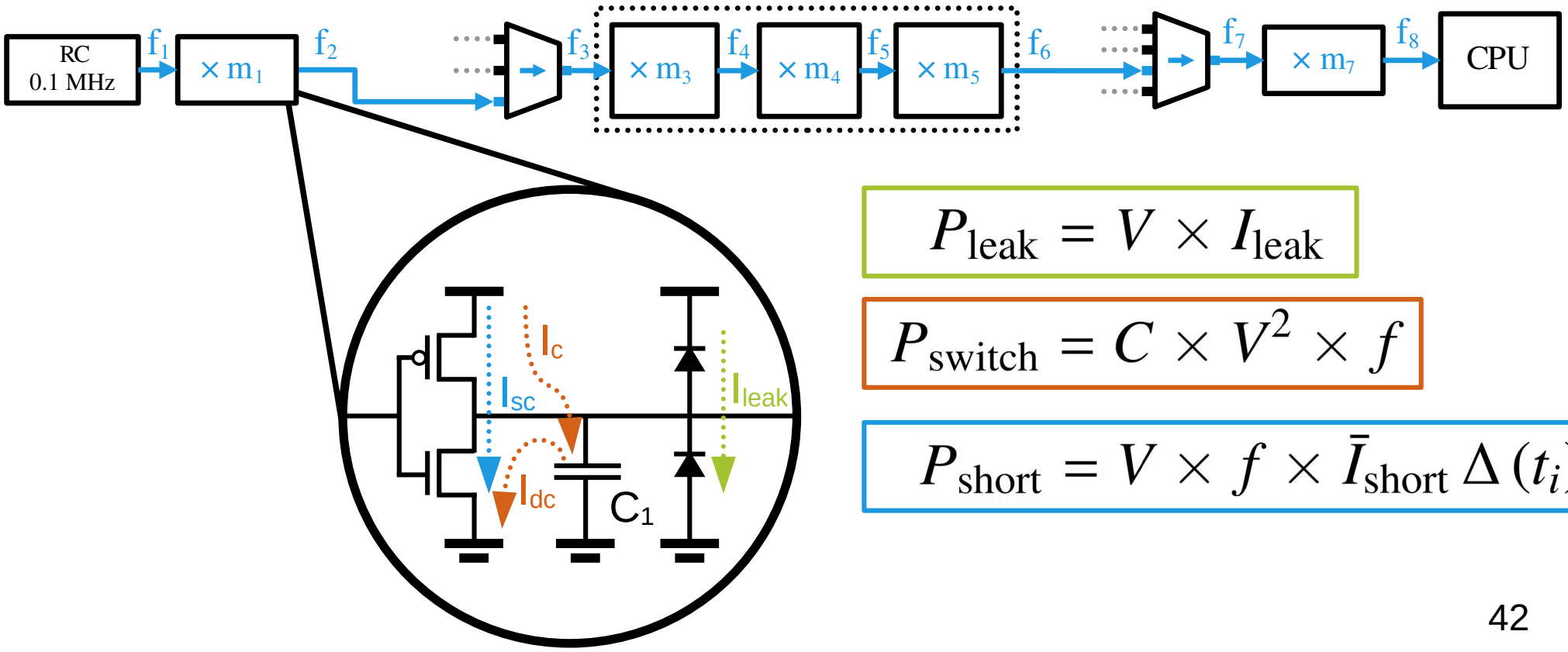
Clock-Tree Power Model: The Idea



Map each scale factor to separate node.



Clock-Tree Power Model: The Idea



$$P_{\text{leak}} = V \times I_{\text{leak}}$$

$$P_{\text{switch}} = C \times V^2 \times f$$

$$P_{\text{short}} = V \times f \times \bar{I}_{\text{short}} \Delta(t_i)$$

Clock-Tree Power Model: The Definition

$$P_{\text{sys-clock}}(\mathcal{N}) = P_{\text{leak}}(\mathcal{N}) + P_{\text{switch}}(\mathcal{N}) + P_{\text{short}}(\mathcal{N})$$

$$= \sum_{i=1}^{k} \left\{ C_{\Gamma(n_i)} \times V^2 \times f_{\Gamma(n_i)} + V \times f_{\Gamma(n_i)} \times \bar{I}_{\text{short}}(\Gamma(n_i))\ \Delta(t_i) + V \times I_{\text{leak}}(\Gamma(n_i)) \right\}$$

$$= \sum_{i=1}^{k} V^2 \left\{ C_{\Gamma(n_i)} \times f_{\Gamma(n_i)} \right\} + V \left\{ f_{\Gamma(n_i)} \times \bar{I}_{\text{short}}(\Gamma(n_i))\ \Delta(t_i) + I_{\text{leak}}(\Gamma(n_i)) \right\}$$

$$P_{\text{sys}}(\mathcal{N}) = P_{\text{sys-stat}}(V) + P_{\text{sys-clock}}(\mathcal{N}) + P_{\text{sys-task}}(V, f)$$

$$\textbf{min}\ \text{SSE} = \sum_{\Gamma} \left(P_{\text{mod}}(\Gamma) - P_{\text{emp}}(\Gamma) \right)^2$$

$$\textbf{s.t.}\ C_{\Gamma(n_i)} \geq 0$$

$$I_{\text{leak}}(\Gamma(n_i)) \geq 0 \quad \text{and} \quad \bar{I}_{\text{short}}(\Gamma(n_i)) \geq 0$$

Clock-Tree Power Model: The Definition

„Summarize consumption of all clock nodes and make them depend on config and hardware properties“

$$P_{\text{sys}}(\mathcal{N}) = P_{\text{sys-stat}}(V) + P_{\text{sys-clock}}(\mathcal{N}) + P_{\text{sys-task}}(V, f)$$

$$\textbf{min}\ \text{SSE} = \sum_{\Gamma} \left(P_{\text{mod}}(\Gamma) - P_{\text{emp}}(\Gamma)\right)^2$$

$$\textbf{s.t.}\ C_{\Gamma(n_i)} \geq 0$$

$$I_{\text{leak}}(\Gamma(n_i)) \geq 0 \quad \text{and} \quad \bar{I}_{\text{short}}(\Gamma(n_i)) \geq 0$$

Clock-Tree Power Model: The Definition

„Summarize consumption of all clock nodes and make them depend on config and hardware properties“

„Relate clock-tree consumption to overall system power“

$$\textbf{min}\ \mathrm{SSE} = \sum_{\Gamma} \left(P_{\mathrm{mod}}(\Gamma) - P_{\mathrm{emp}}(\Gamma) \right)^2$$

$$\textbf{s.t.}\ C_{\Gamma(n_i)} \geq 0$$

$$I_{\mathrm{leak}}(\Gamma(n_i)) \geq 0 \quad \text{and} \quad \bar{I}_{\mathrm{short}}(\Gamma(n_i)) \geq 0$$

Clock-Tree Power Model: The Definition

„Summarize consumption of all clock nodes and make them depend on config and hardware properties“

„Relate clock-tree consumption to overall system power“

„Optimization problem for extracting hardware properties from empirical measurement“

$$\textbf{s.t.}\ C_{\Gamma(n_i)} \geq 0$$

$$I_{\text{leak}}(\Gamma(n_i)) \geq 0 \quad \text{and} \quad \bar{I}_{\text{short}}(\Gamma(n_i)) \geq 0$$

Clock-Tree Power Model: The Definition

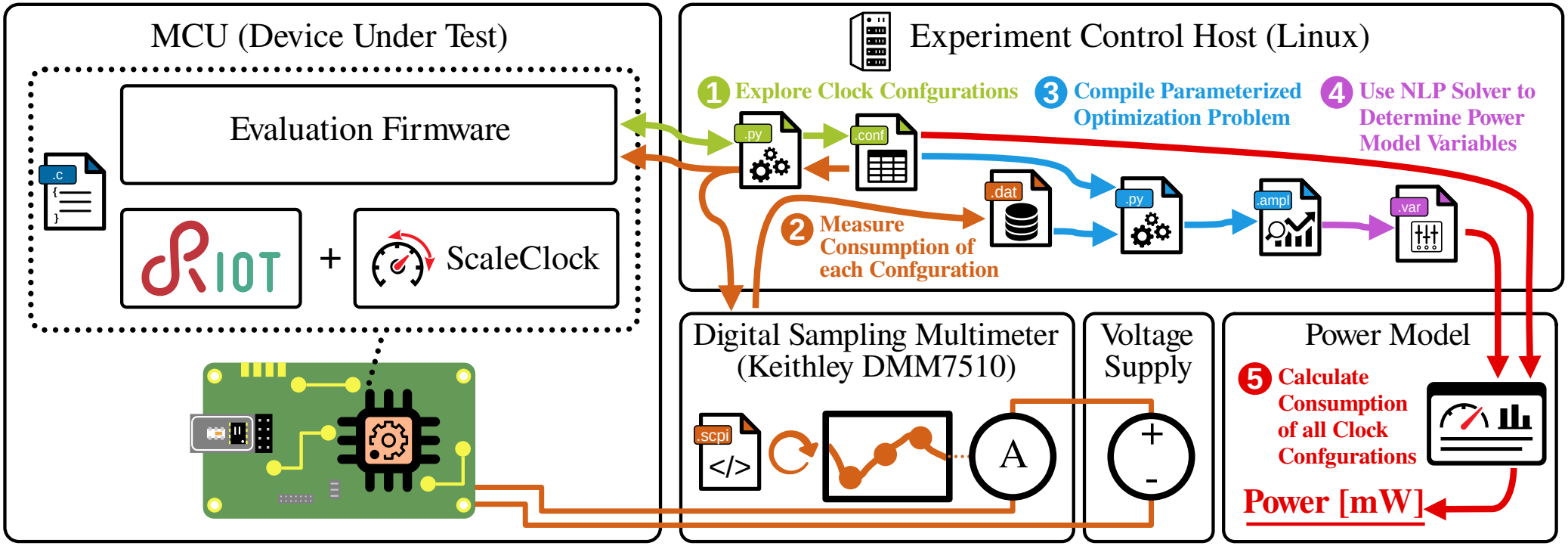
„Summarize consumption of all clock nodes and make them depend on config and hardware properties“

„Relate clock-tree consumption to overall system power“

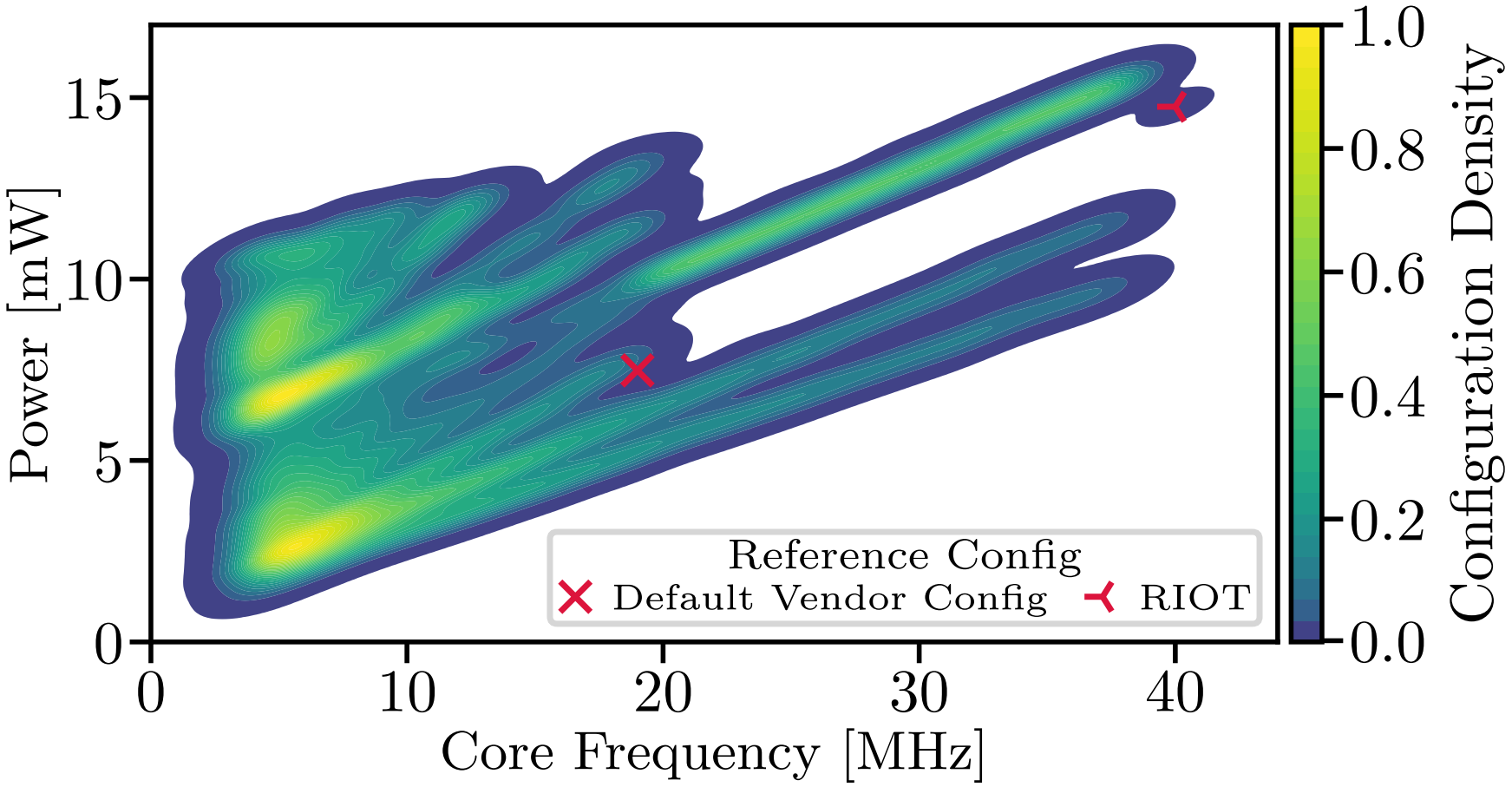
„Optimization problem for extracting hardware properties from empirical measurement“

„Constraints to ensure the solution makes sense physically“

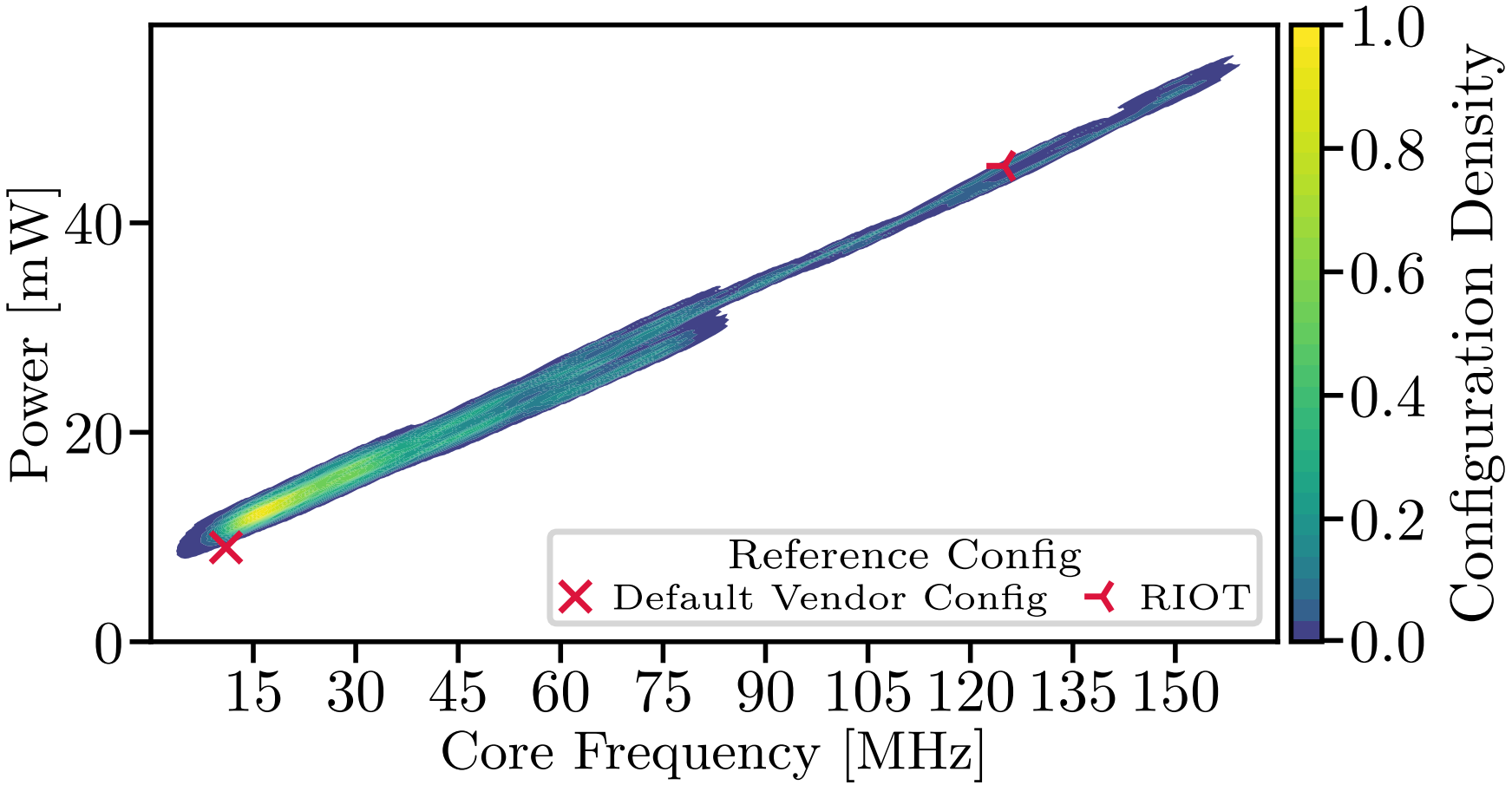
Clock-Tree Power Model: The Parametrization



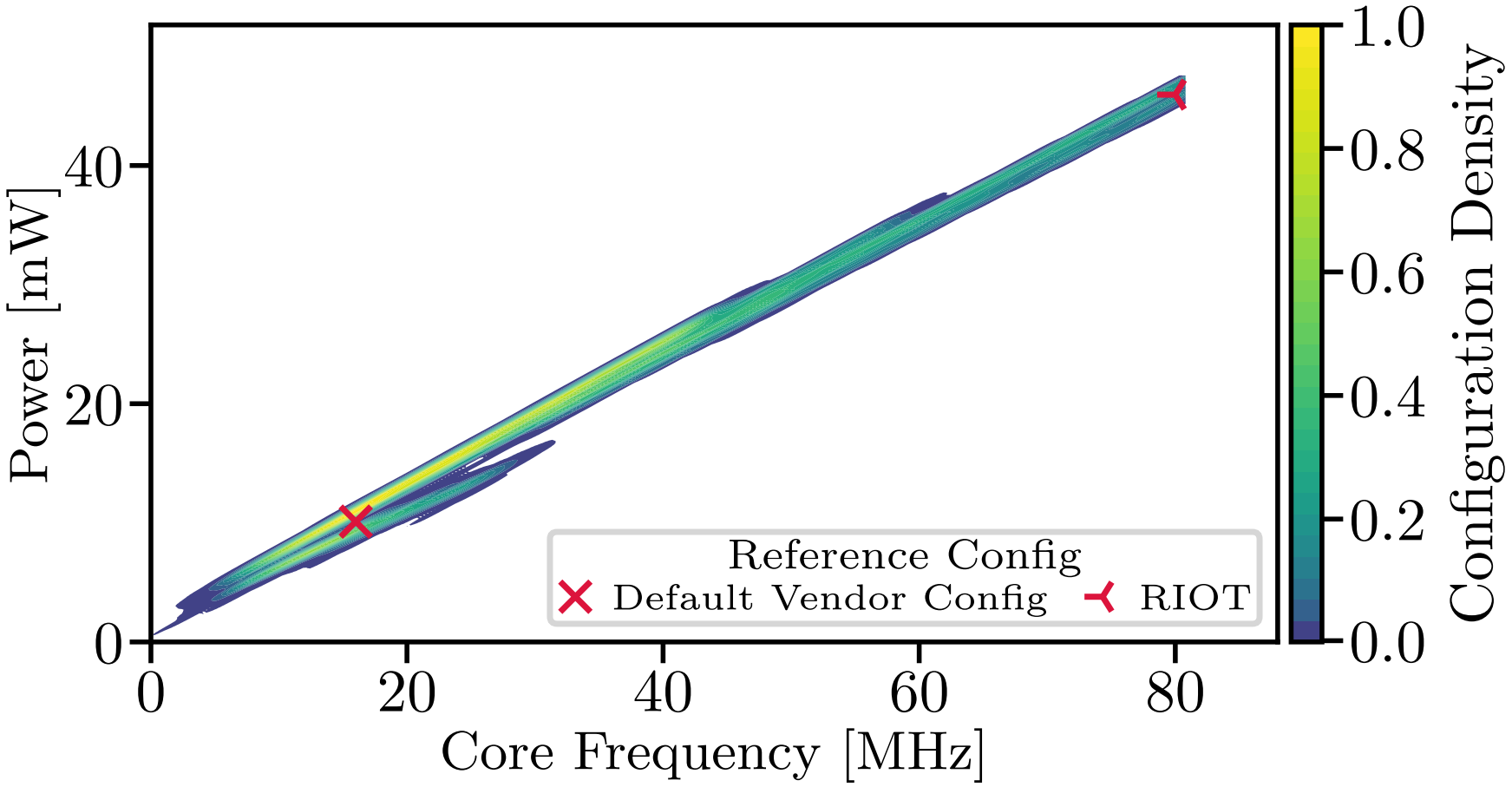
Configuration-Space (slstk3402a)



Configuration-Space (rpi-pico-2-riscv)

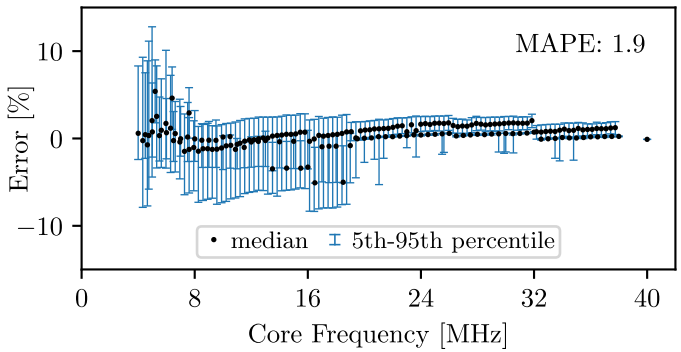


Configuration-Space (nucleo-l476rg)

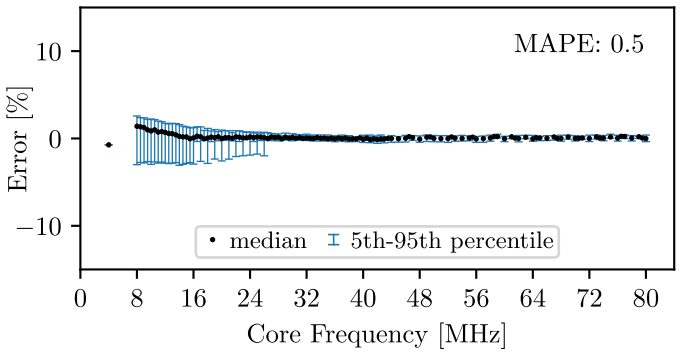


Power Model Accuracy

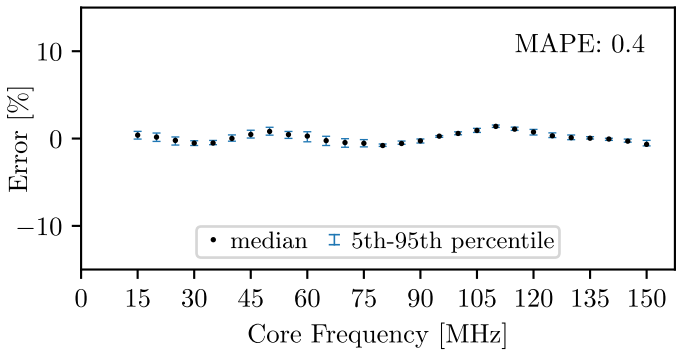
slstk3402a



nucleo-l476rg



rpi-pico-2-riscv



- Model generalizes across manufacturers and core architectures
  - Configuration-space is covered with small errors

Conclusion and Outlook

Conclusion

- DVFS enables self-adaptive IoT devices with better energy-awareness
  - Can save significant energy on top of plain duty-cycling
- Core frequency alone only tells part of the story
  - Selection of intermediate scale factors matters!
  - Relevant also for static scenarios!
- Clock-tree power model based on physical surrogate circuits
  - accurately predicts the power consumption for arbitrary configurations
  - covers the full configuration space
  - more comprehensive than (cherry-picked?) datasheet values

Outlook

- Similar formalized methods needed ...
  - For transition timing
  - For (hardware specific) software scalability
- Experiment study with representative set of software modules
- Follow the vision of ”`USEMODULE += auto_dvfs`“

Questions?

Michel Rottleuthner, Thomas C. Schmidt, Matthias Wählisch,
A Power Model of System Clocks to Minimize Energy Consumption on Constrained MCUs,
IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), IEEE : Piscataway, NJ, USA, September 2026, (in press).

Michel Rottleuthner, Thomas C. Schmidt, Matthias Wählisch,
Duty-Cycling is Not Enough in Constrained IoT Networking: Revealing the Energy Savings of Dynamic Clock Scaling,
Technical Report, No. 2508.09620, August 2025.

Michel Rottleuthner, Thomas C. Schmidt, Matthias Wählisch,
Dynamic Clock Reconfiguration for the Constrained IoT and its Application to Energy-efficient Networking,
In: Proc. of Embedded Wireless Systems and Networks (EWSN'22), p. 168–179, ACM : New York, USA, October 2022.