The TURBO Diaries: Application-based Frequency Scaling Explained

Jons-Tobias Wamhoff
Stephan Diestelhorst
Christof Fetzer
Technische Universität Dresden, Germany

Patrick Marlier
Pascal Felber
Université de Neuchâtel, Switzerland

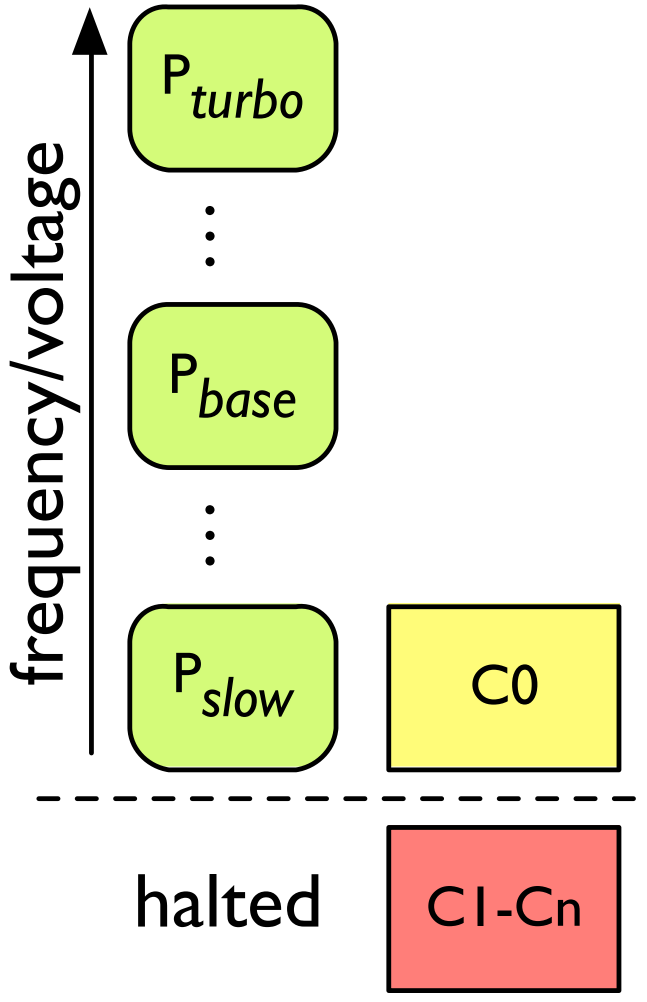
Dave Dice
Oracle Labs, USA

Overview

- Dynamic voltage and frequency scaling (DVFS)
  - traditionally: used to *save energy* or *boost* sequential bottlenecks/serial peak loads
  - today: improve performance by exposing asymmetric properties of applications

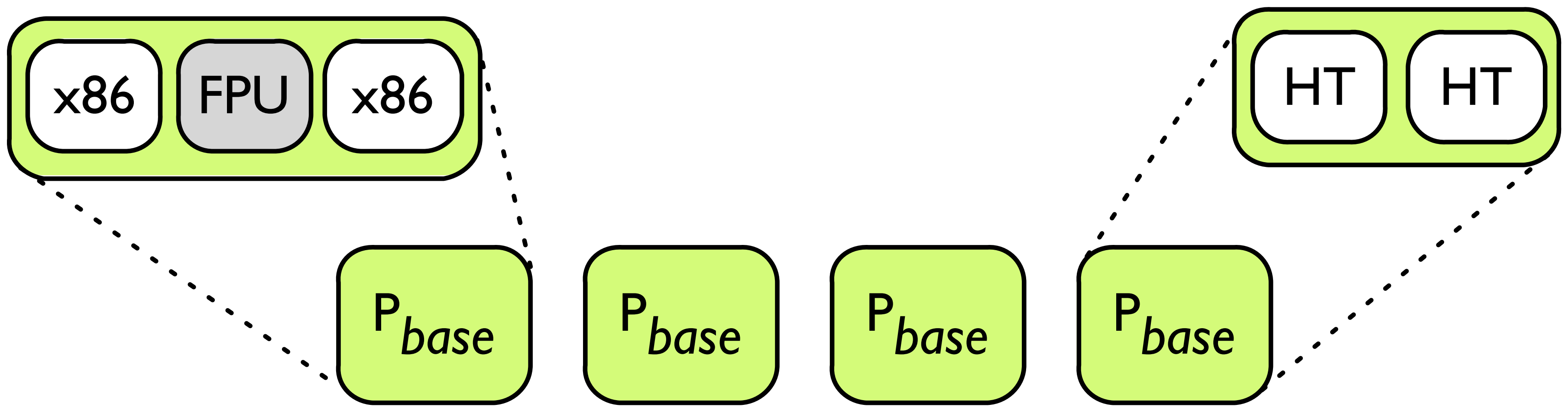
- Outline
  - Recap DVFS features on current x86 multicores
  - DVFS properties: latency and power
  - Applying DVFS on application-level

P- and C-states

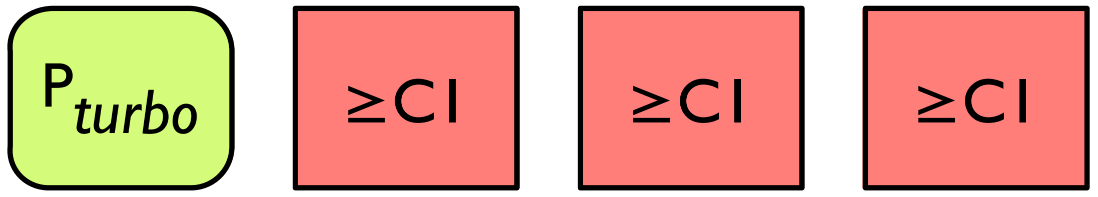


- P-states: performance states
  - predefined frequency/voltage pairs
  - controlled through machine-specific registers (MSRs, privileged rdmsr/wrmsr)
- C-states: power states
  - trade entry/wakeup latency for higher power savings
  - entered by hlt or monitor/mwait

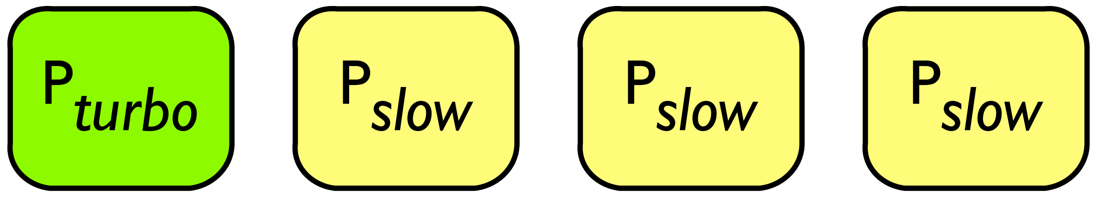
AMD Turbo CORE & Intel Turbo Boost



- Voltage and frequency domain: *module* vs. *package*

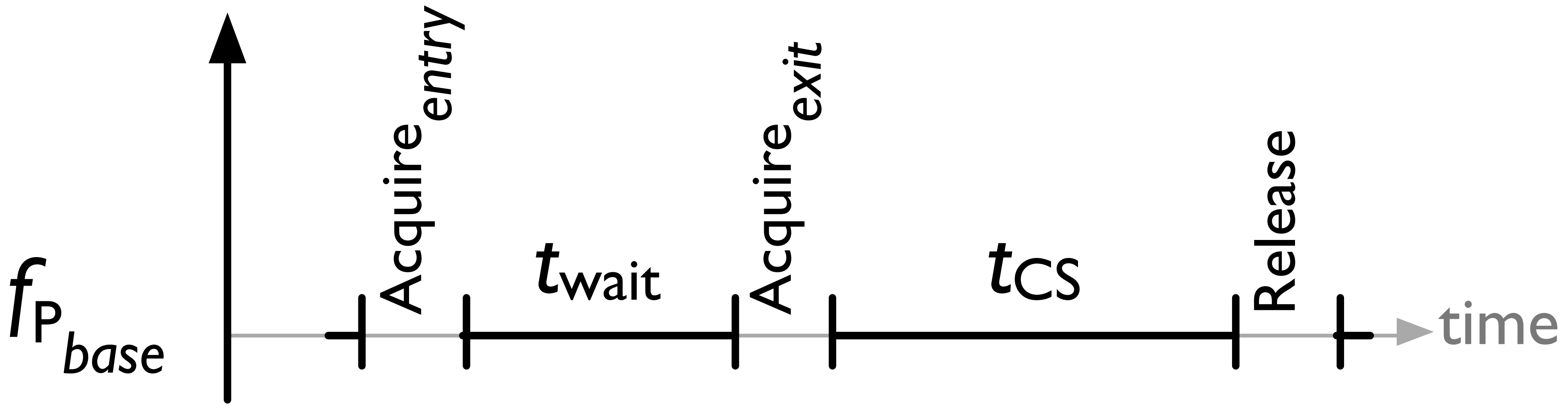


- Boosting: *deterministic* vs. *thermal*



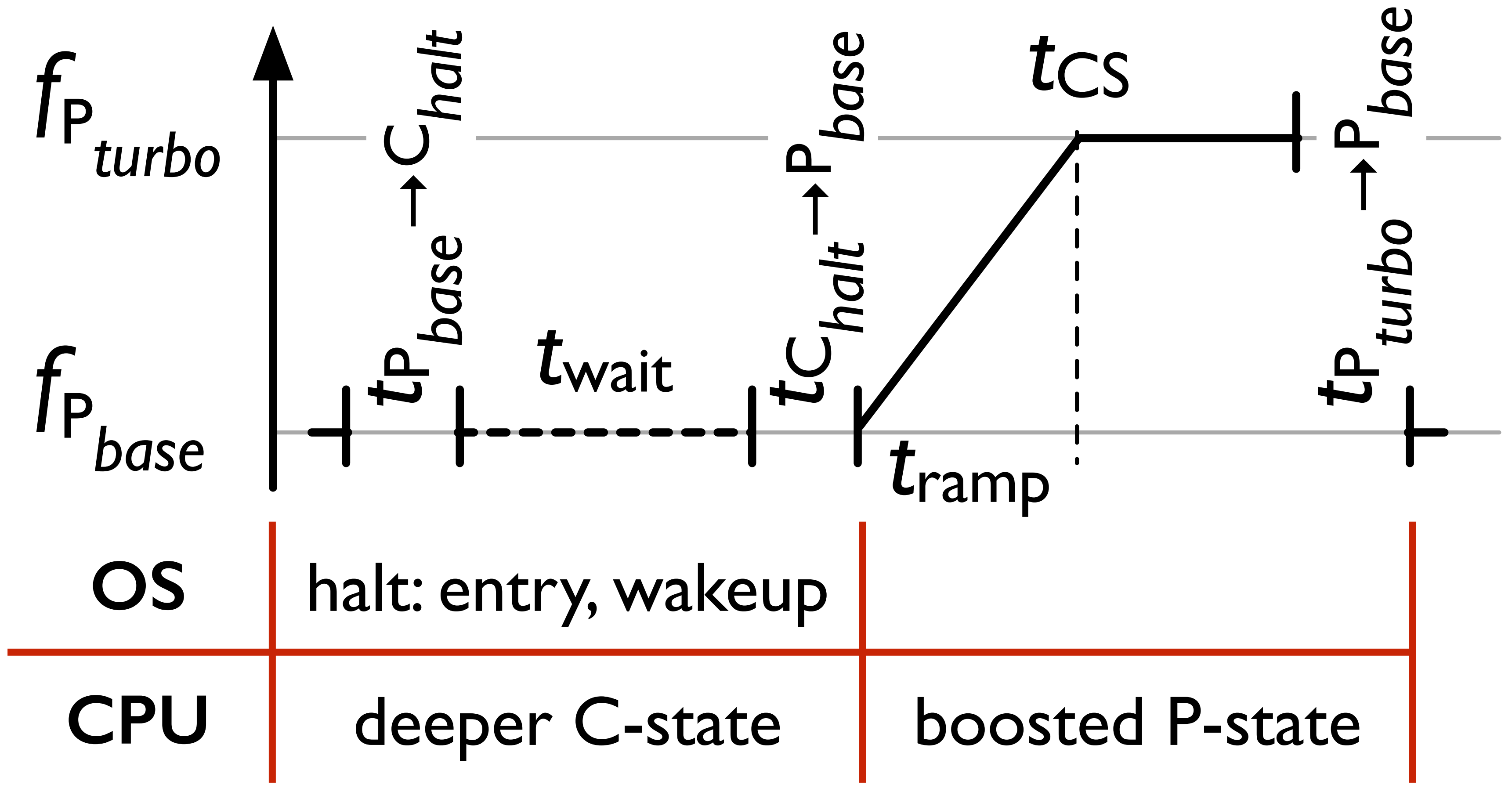
- AMD only: asymmetric frequencies with manual boost

Evaluation Setup



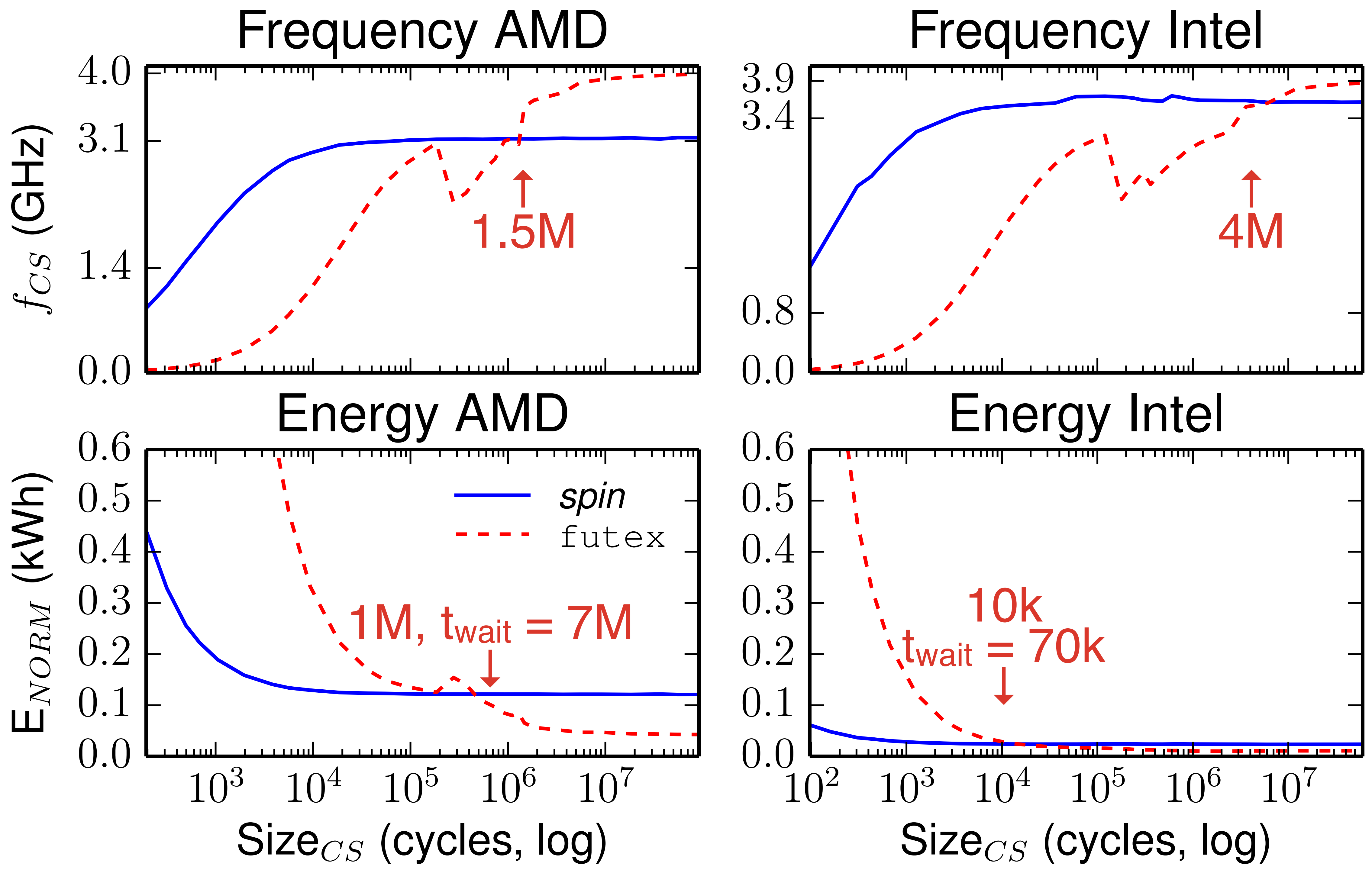
- Critical sections (CS) protected by MCS queue lock
  - *Decorations* on acquire/release → *trigger* DVFS
  - Variable *size* of CS → *amortize* DVFS cost
  - Effective CS *frequency*: $f_{CS} = f_{base} \cdot \frac{t_{CS}}{t_{A+CS+R}}$
  - *Energy* for 1 hour at P_{base}: $E_{NORM} = E_{sample} \cdot \frac{t_{A+CS+R}}{t_{CS}}$

Automatic Frequency Scaling

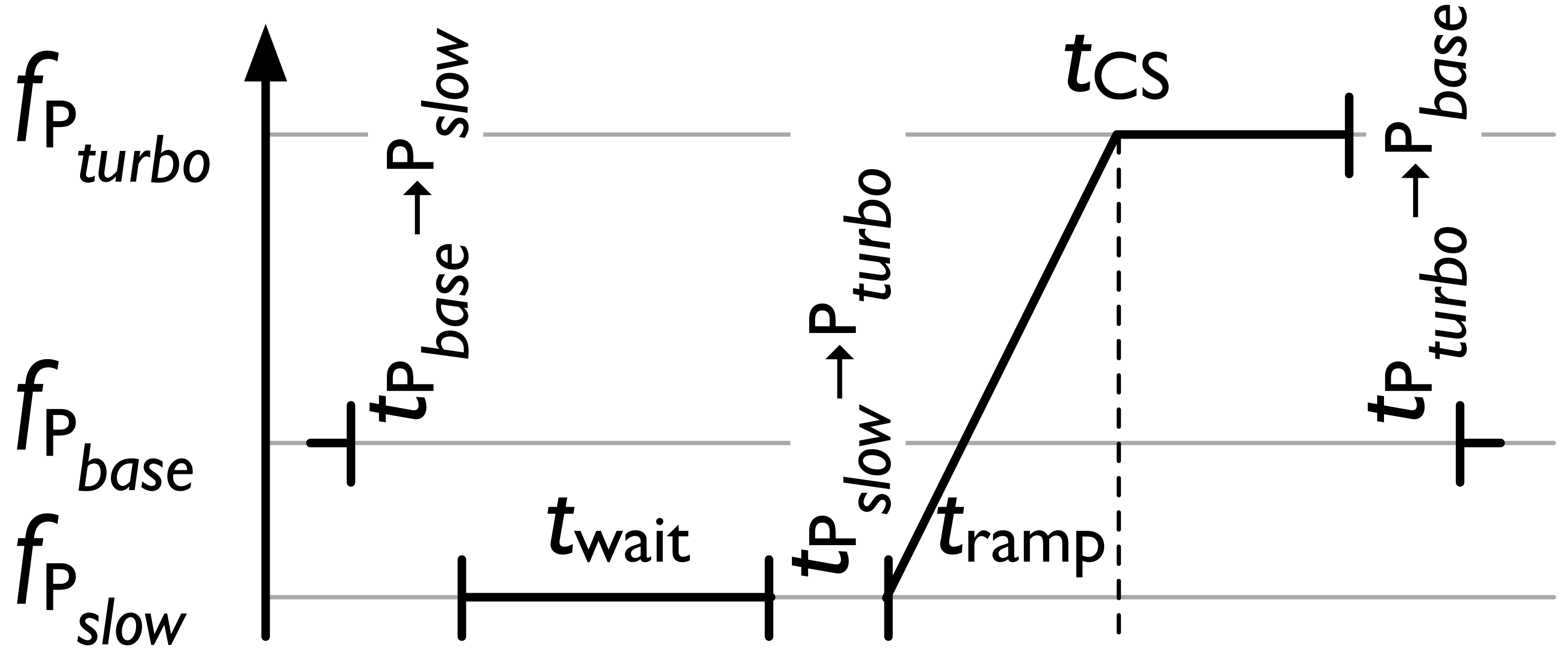


- Decoration: *spinning vs. blocking*
- P-state transitions triggered by hardware

Blocking vs. Spinning Locks



Manual Frequency Scaling

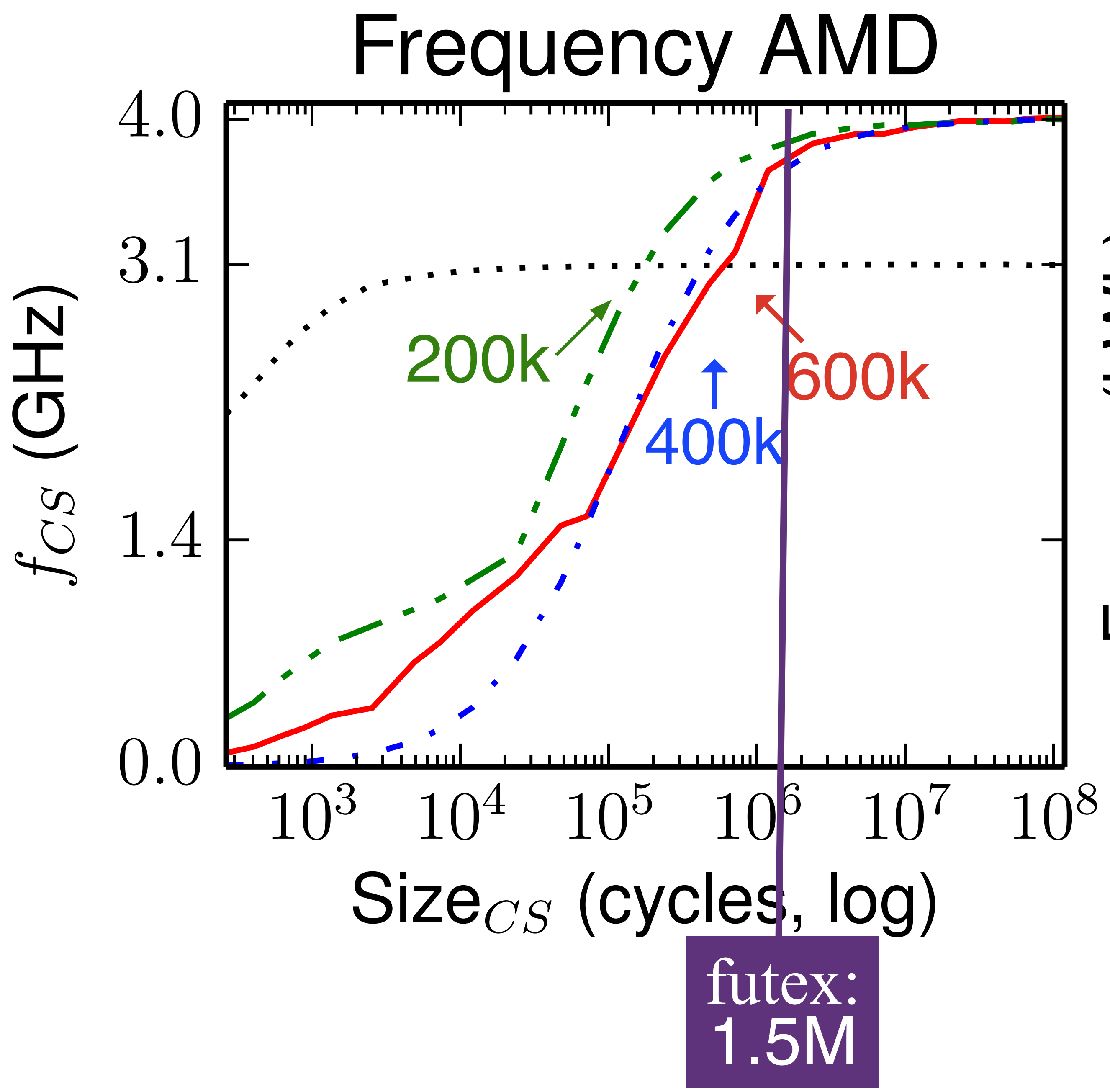


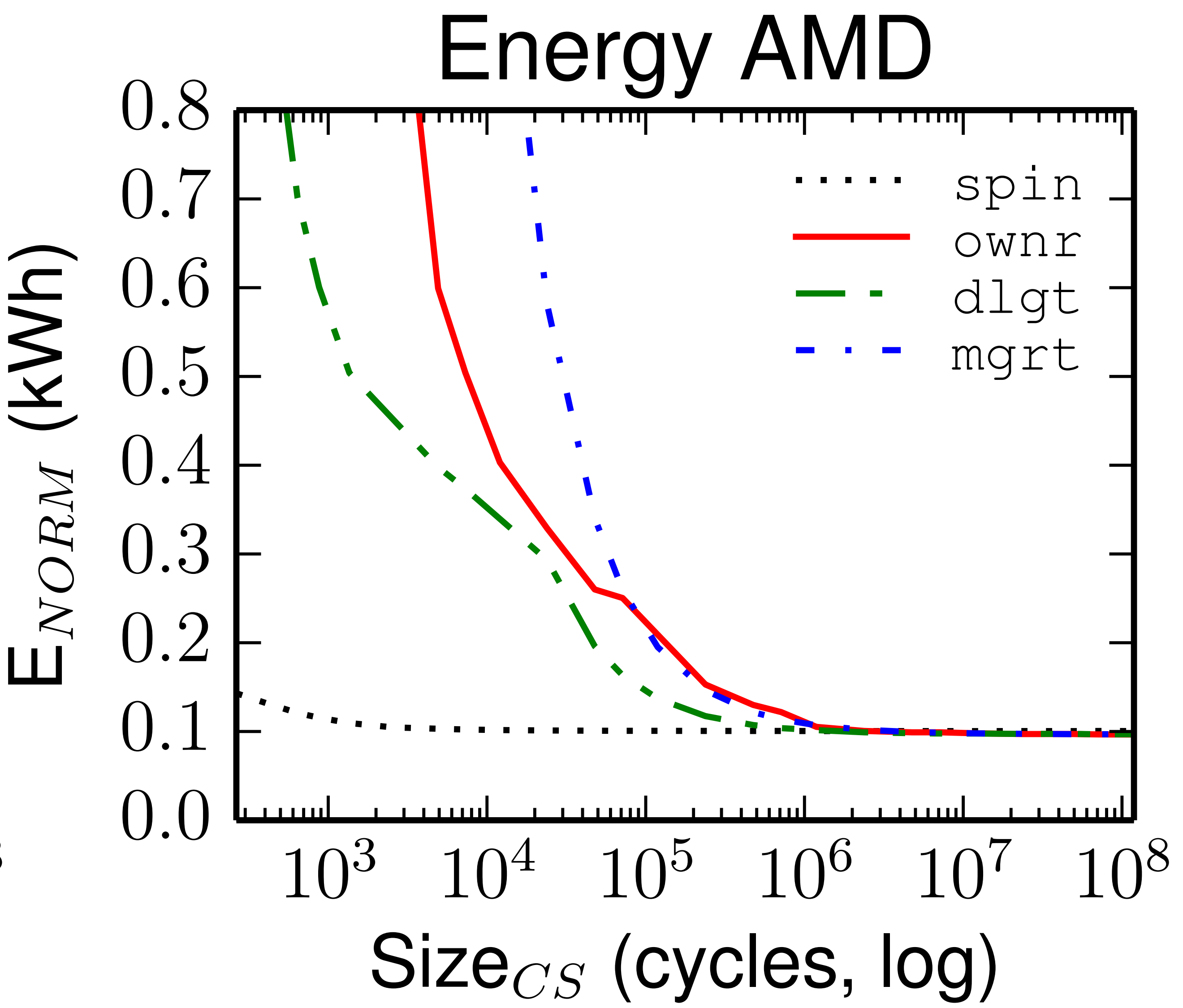
$f_{P_{turbo}}$
$f_{P_{base}}$
$f_{P_{slow}}$
$t_{P_{base} \to P_{slow}}$
t_{wait}
$t_{P_{slow} \to P_{turbo}}$
t_{ramp}
t_{CS}
$t_{P_{turbo} \to P_{base}}$

ioctl	1k		1k			1k
wrmsr	28k		2k			23k
transition	2k			225k		1k

- Decoration: *spin* and application-level DVFS control

Manual Lock Boosting

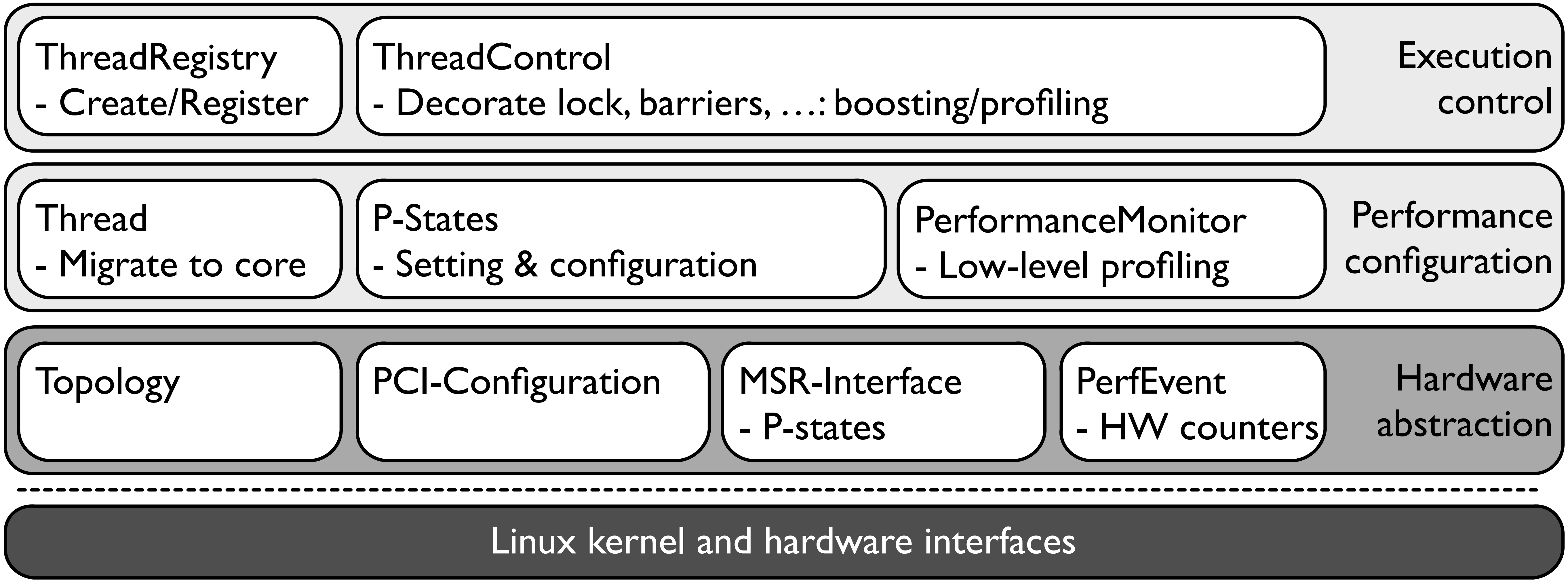




- *spin*: static P_{base}
- *owner*: dynamically boost
- *delegate*: dedicated wrmsr core
- *migrate*: statically boosted core

TURBO Library

- Convenient programmatical application-level DVFS control
- Testbed to explore challenges of future heterogeneous cores



https://bitbucket.org/donjonsn/turbo

Boosting Applications

- *Expose application knowledge*
  - Asymmetric software transactional memory: up to 50% speedup with only 2% more energy
- *Tradeoffs* when IPC depends on core frequency
  - Hash table resize in memcached: 9% speedup but 22% higher frequency
- Outweigh P-state latency by *delegating* CS
  - High cross-module round-trip delay (2k cycles)
  - Intra-module delay scales with P-state (P_{boost}: 280 cycles)