

Hannes Tröpgen¹, Till Smejkal², Thomas Ilsche¹, Robert Schöne¹, Horst Schirmeier²

¹ ZIH, CIDS, TU Dresden

² Chair of Operating Systems, Faculty of CS, TU Dresden

Pinpointing Idle-Power Regressions in Linux

EESP Workshop 2025

Hamburg, Germany // 13 June 2025

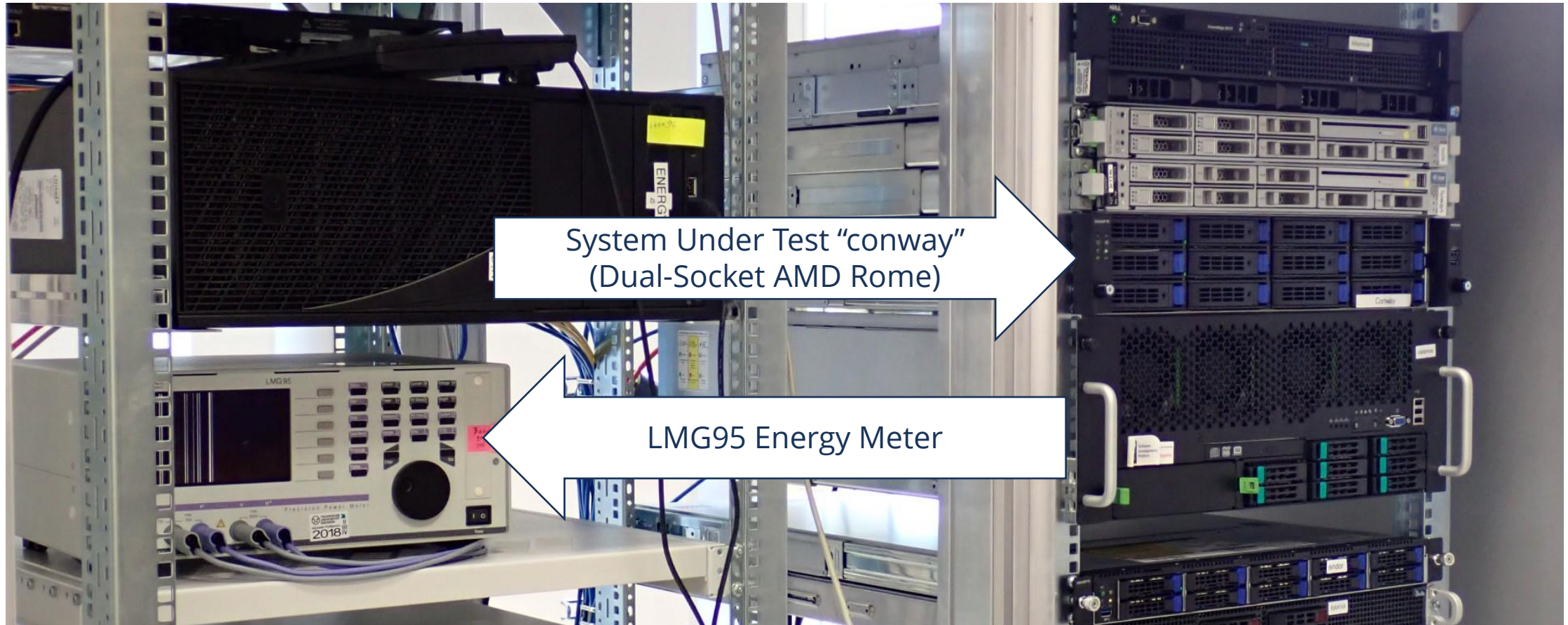


Motivation



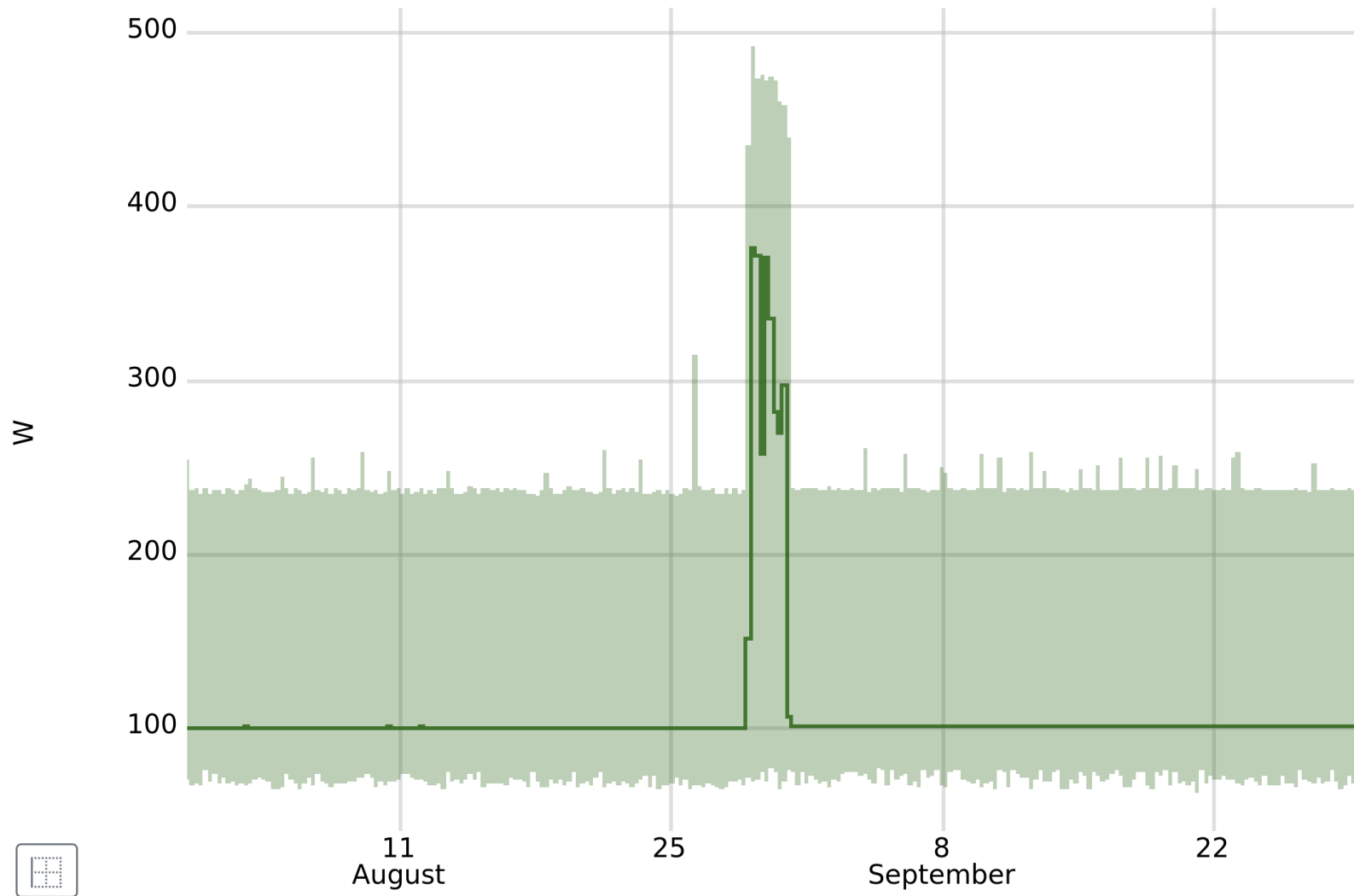
Racks in Energy Lab @ TUD

Motivation



Racks in Energy Lab @ TUD

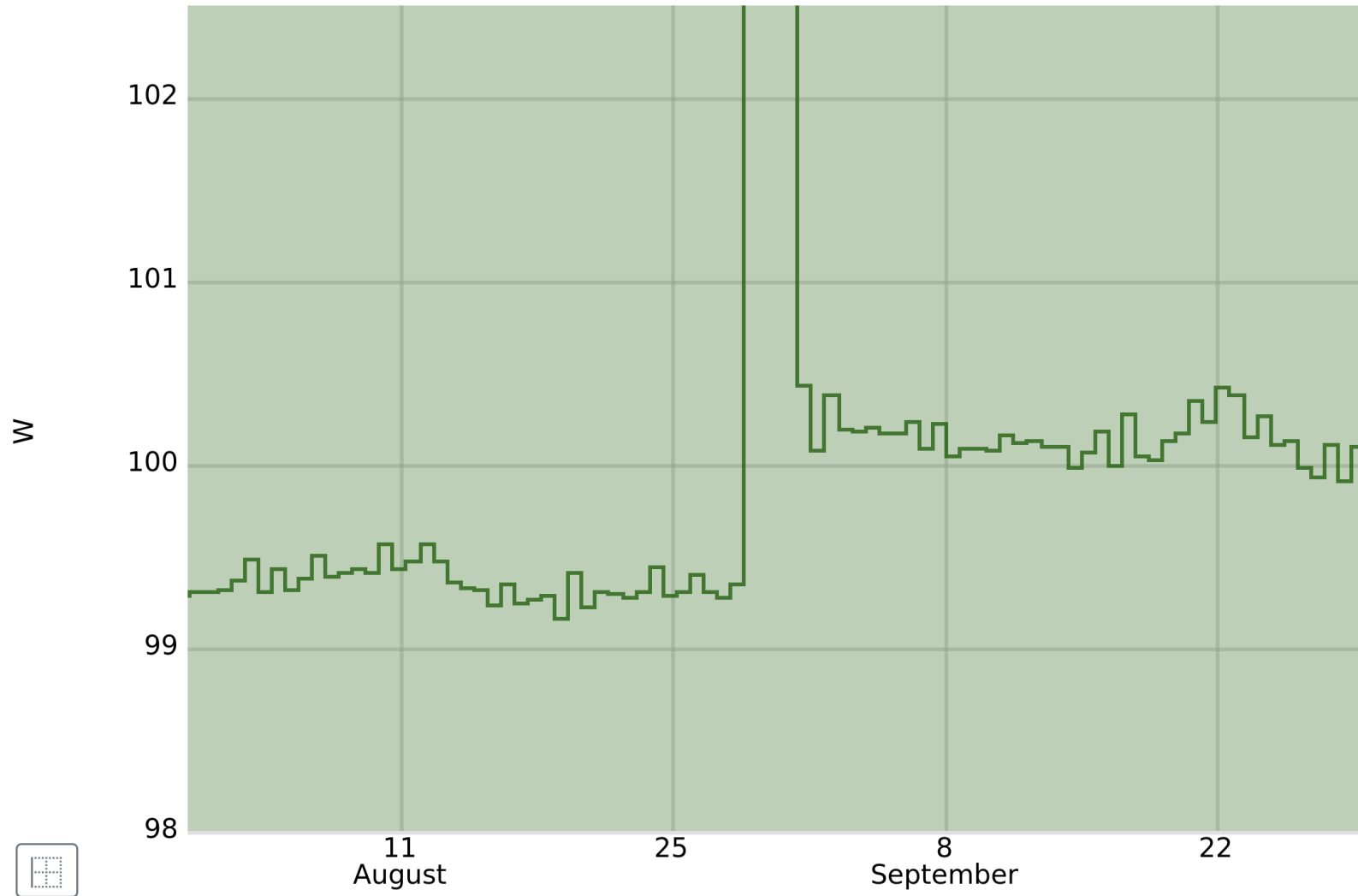
Motivation



Power consumption of conway as reported by energy meter, August & September 2022

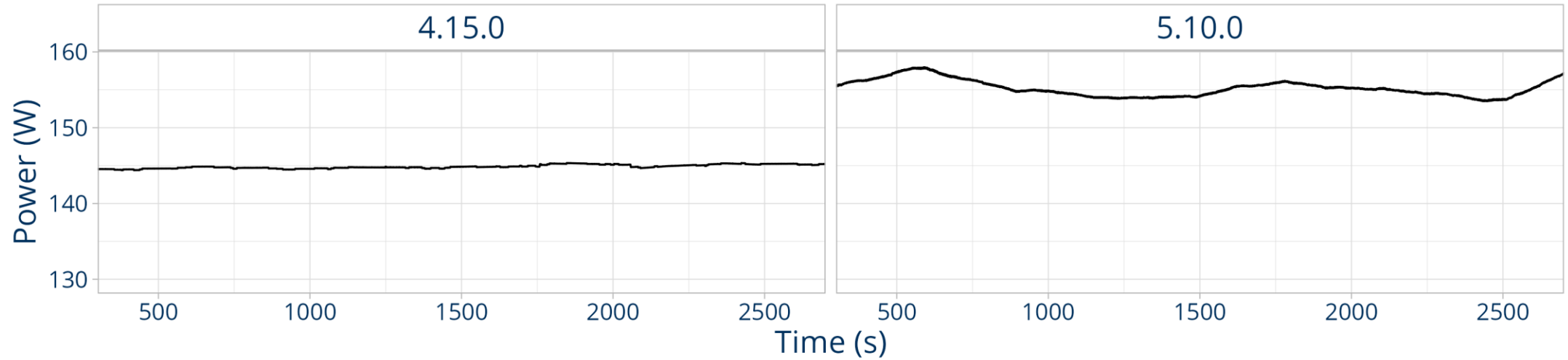


Motivation



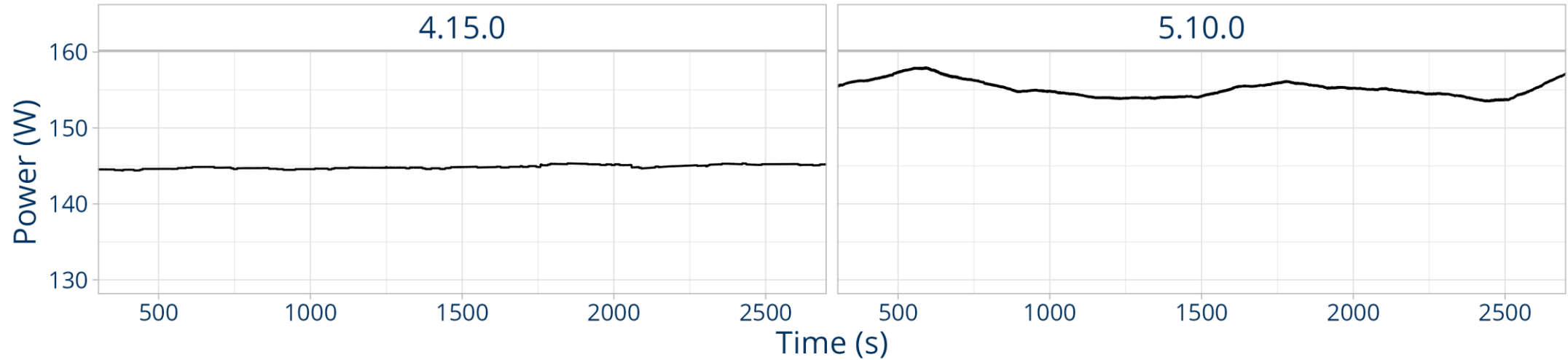
Power consumption of conway
as reported by energy meter,
August & September 2022

Motivation



Power consumption (rollmean across 300s) during idle for different Linux kernels on conway

Motivation



Power consumption (rollmean across 300s) during idle for different Linux kernels on conway

- Can we find the commit that caused this regression?
- Develop general process to pinpoint energy regressions to commits
- Avoid probe effect

Bisection

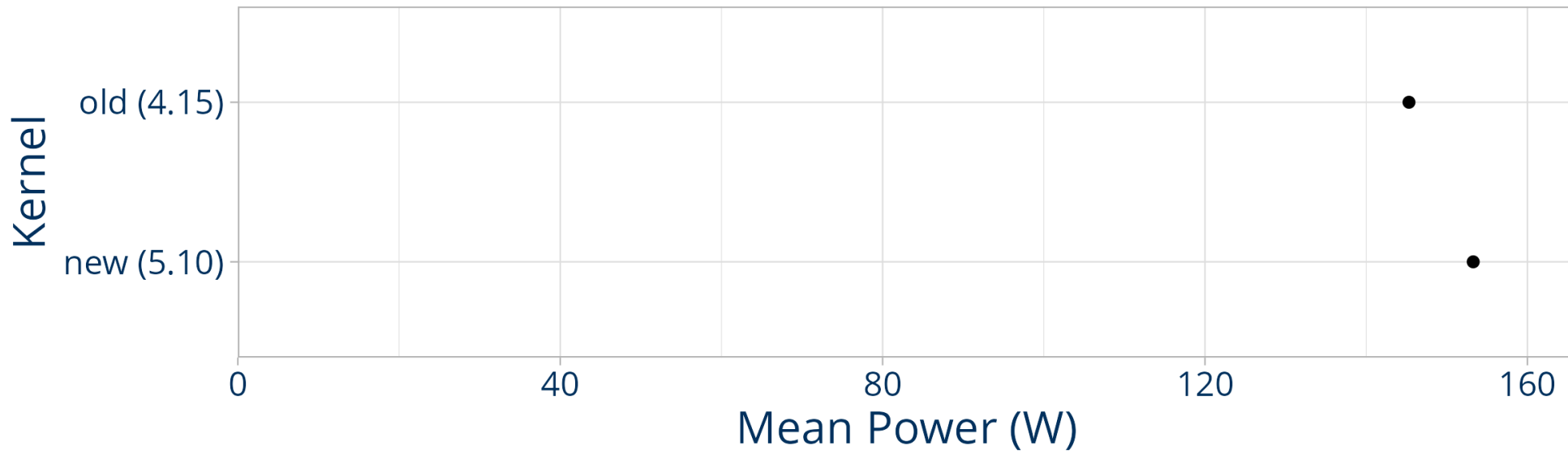
- Scalable algorithm to find regression in history of commits
- Start with known-good & known-bad commit
- Repeat:
 - Version control system (git) selects commit to be classified
 - User classifies commit as good/bad (also: old/new)
- Final Result: first new commit

Classification: Reference Measurements

- Classify by comparing to reference cases
- Naive: difference between mean power consumption
 - Can be selected depending on use case

Classification: Reference Measurements

- Classify by comparing to reference cases
- Naive: difference between mean power consumption
 - Can be selected depending on use case



Mean power consumption of reference measurements (idle)

Bisection Result

First new commit after 17 steps: cea79e7e2f24 "apei/ghes: Do not delay GHES polling"

```
diff --git a/drivers/acpi/apei/ghes.c b/drivers/acpi/apei/ghes.c
index 8906c80175e684..103acbbfcf9a51 100644
--- a/drivers/acpi/apei/ghes.c
+++ b/drivers/acpi/apei/ghes.c
@@ -1180,7 +1180,7 @@ static int ghes_probe(struct platform_device *ghes_dev)

     switch (generic->notify.type) {
     case ACPI_HEST_NOTIFY_POLLED:
-        timer_setup(&ghes->timer, ghes_poll_func, TIMER_DEFERRABLE);
+        timer_setup(&ghes->timer, ghes_poll_func, 0);
         ghes_add_timer(ghes);
         break;
     case ACPI_HEST_NOTIFY_EXTERNAL:
```

Example Case 2: Powernightmares

- Powernightmares [1]: sub-optimal sleep-state (“C-state”) selection during idle
 - Requires specific activity pattern, can be artificially triggered
 - Fixed in Linux 4.17



Sleep states during a Powernightmare (active [blue], C1E [green], C6 [red]), Fig. 1a from [1]

Example Case 2: Powernightmares

- Powernightmares [1]: sub-optimal sleep-state (“C-state”) selection during idle
 - Requires specific activity pattern, can be artificially triggered
 - Fixed in Linux 4.17



Sleep states during a Powernightmare (active [blue], C1E [green], C6 [red]), Fig. 1a from [1]

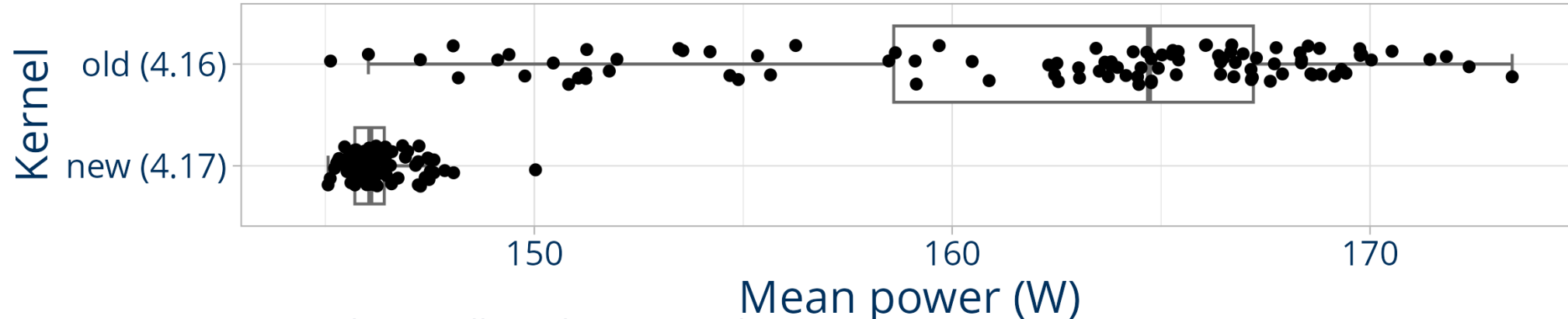
- Goal: Find fix of Powernightmares using bisection
 - Use artificial trigger to enhance reproducibility
 - Old: with Powernightmares (more energy used)
 - New: Powernightmares fixed (less energy used)

Robustness: Repetitions

- Successful bisection requires correct classification at every bisection step
 - Assume 99% overall accuracy for 14 step-bisection
 - 99.9% accuracy per step required

Robustness: Repetitions

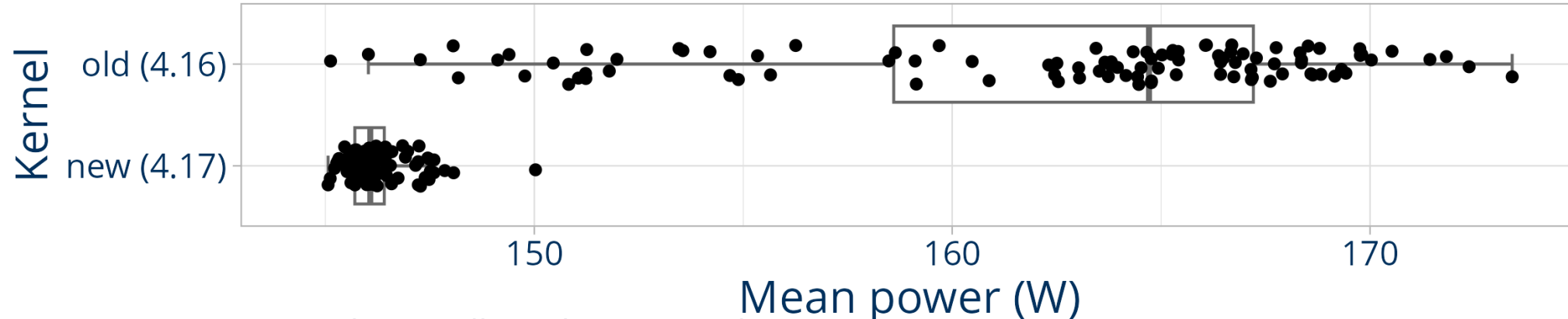
- Successful bisection requires correct classification at every bisection step
 - Assume 99% overall accuracy for 14 step-bisection
 - 99.9% accuracy per step required



Power consumption during idle with powernightmare trigger

Robustness: Repetitions

- Successful bisection requires correct classification at every bisection step
 - Assume 99% overall accuracy for 14 step-bisection
 - 99.9% accuracy per step required



Power consumption during idle with powernightmare trigger

- Repeat measurements
 - Including reference measurements
 - Adjust classification function

Robustness: Verification Measurements

- Examine found *first new commit* in isolation
- Show that *first new commit* is necessary & sufficient to cause behavior change
 - Sufficient: take old reference kernel, apply *first new commit*, must classify as new
 - Necessary: take new reference kernel, revert *first new commit*, must classify as old
- Might not apply/revert cleanly, approaches:
 - Apply/revert commits as group
 - Fix errors manually
 - Skip & and move on

Bisection: Powernightmares

First new commit 554c8aa8ecad found by powernightmare bisection

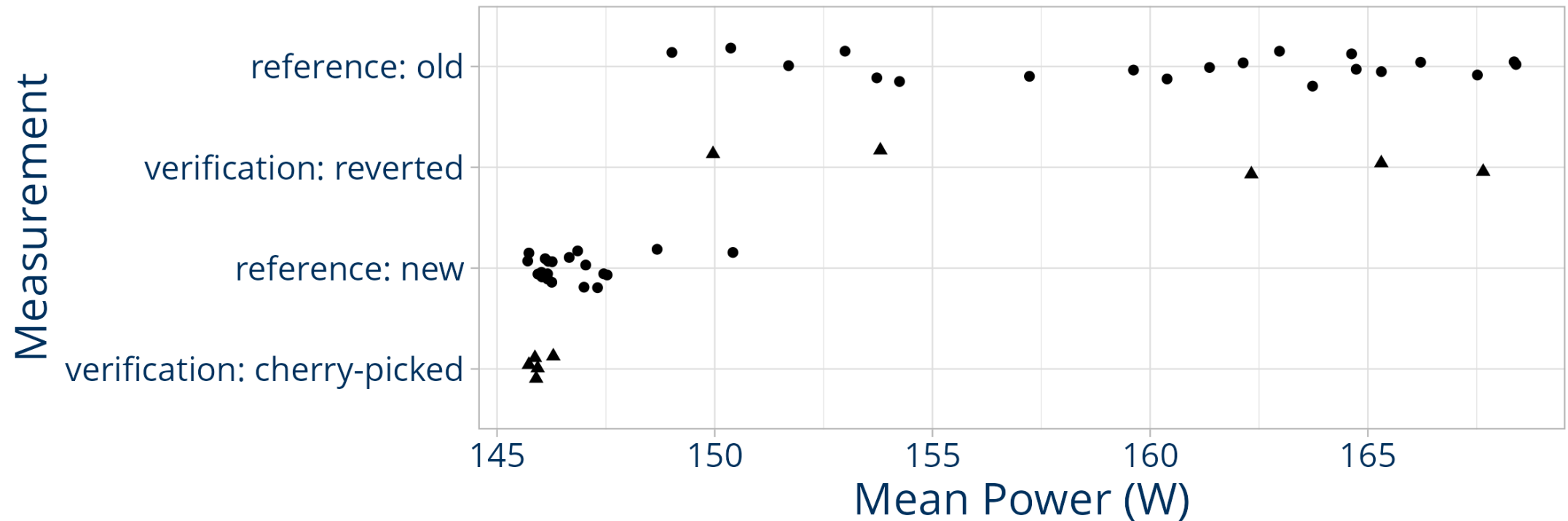
author Rafael J. Wysocki <rafael.j.wysocki@intel.com> 2018-04-03 23:17:11 +0200
committer Rafael J. Wysocki <rafael.j.wysocki@intel.com> 2018-04-09 11:54:07 +0200
commit [554c8aa8ecade210d58a252173bb8f2106552a44](#) (patch)
tree [6712ac8a8c4ccf95730c1c1c4cdafe595280a578](#)
parent [a59855cd8c613ba4bb95147f6176360d95f75e60](#) (diff)
download [linux-554c8aa8ecad.tar.gz](#)

sched: idle: Select idle state before stopping the tick

In order to address the issue with short idle duration predictions by the idle governor after the scheduler tick has been stopped, reorder the code in `cpuidle_idle_call()` so that the governor idle state selection runs before `tick_nohz_idle_go_idle()` and use the "nohz" hint returned by `cpuidle_select()` to decide whether or not to stop the tick.

This isn't straightforward, because `menu_select()` invokes

Verification Measurements: Powernightmares



• reference ▲ verification

Summary

- Bisection driven by power measurements
- Classify by comparing to reference measurements
- Sensitivity to errors mitigated by
 - Measurement repetition
 - Verification Measurements
- Demonstrated on two examples

Future Work

- Fuse setup into single machine
 - Evaluate RAPL & probe effect
- Alternatives to bisection algorithm
- Evaluate alternative classification methods

Thanks for your attention
Questions?

References

[1] Ilsche, T., Hähnel, M., Schöne, R., Bielert, M., Hackenberg, D. (2018). Powernightmares: The Challenge of Efficiently Using Sleep States on Multi-core Systems. In: Heras, D., *et al.* Euro-Par 2017: Parallel Processing Workshops. Euro-Par 2017. Lecture Notes in Computer Science(), vol 10659. Springer, Cham.
https://doi.org/10.1007/978-3-319-75178-8_50

Backup Slides

Infrastructure

