university of groningen

faculty of science and engineering

Container-Level Energy Observability in Kubernetes Clusters

ICT4S 2025
Dublin, Ireland

Bjorn Pijnacker[1]
Author & Speaker

Vasilios Andrikopoulos[1]
Co-author

Brian Setz[1]
Co-author

[1] University of Groningen

Introduction

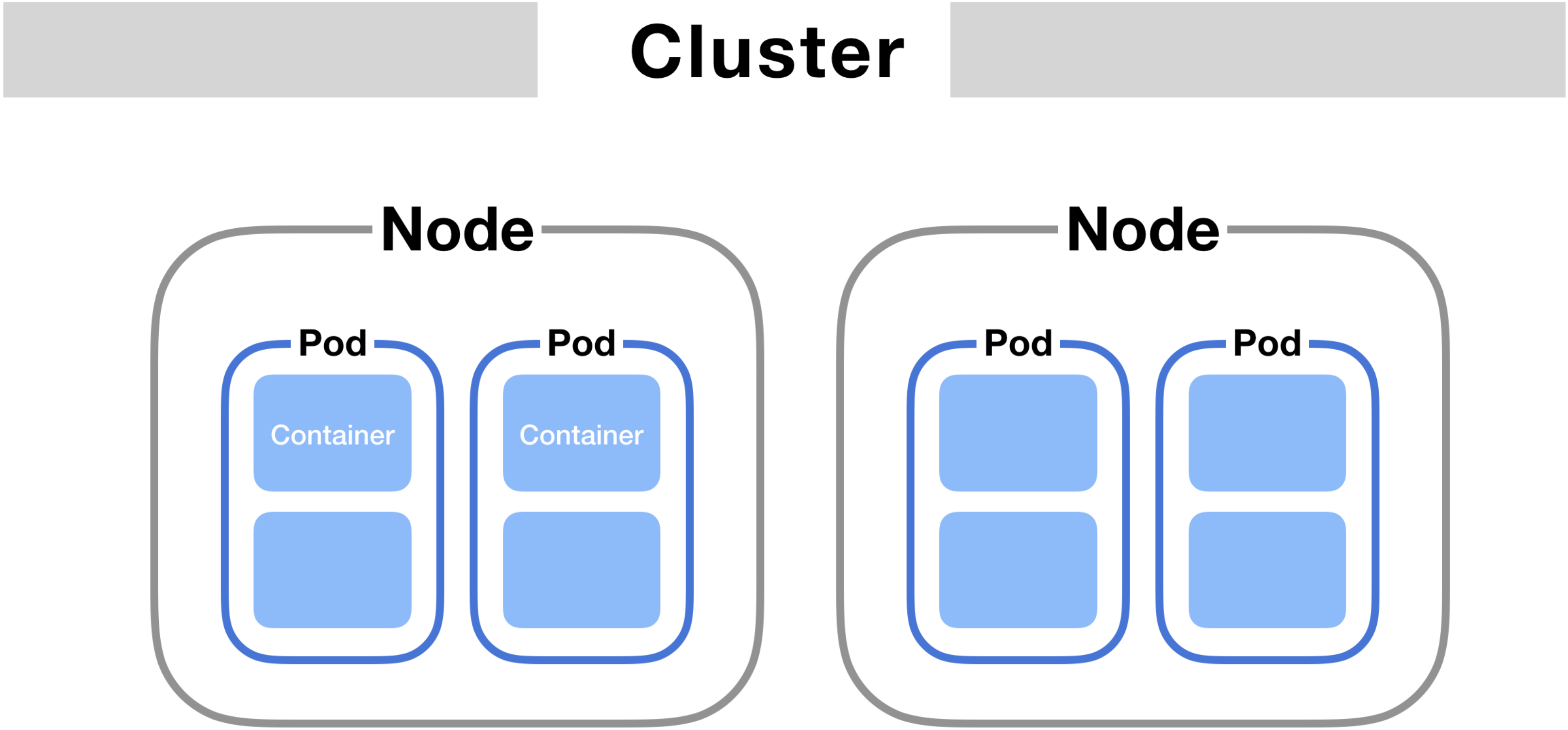
- Research in energy efficiency focus on **efficiency/waste** and reducing **overprovisioning**
- Existing solutions:
  - Optimize entire cluster or datacenter at once, little regard for the individual workload
- Missing: insight (observability) into workload power usage in Kubernetes

How can we accurately measure (or estimate) the power usage of Kubernetes containers based on external measurements?

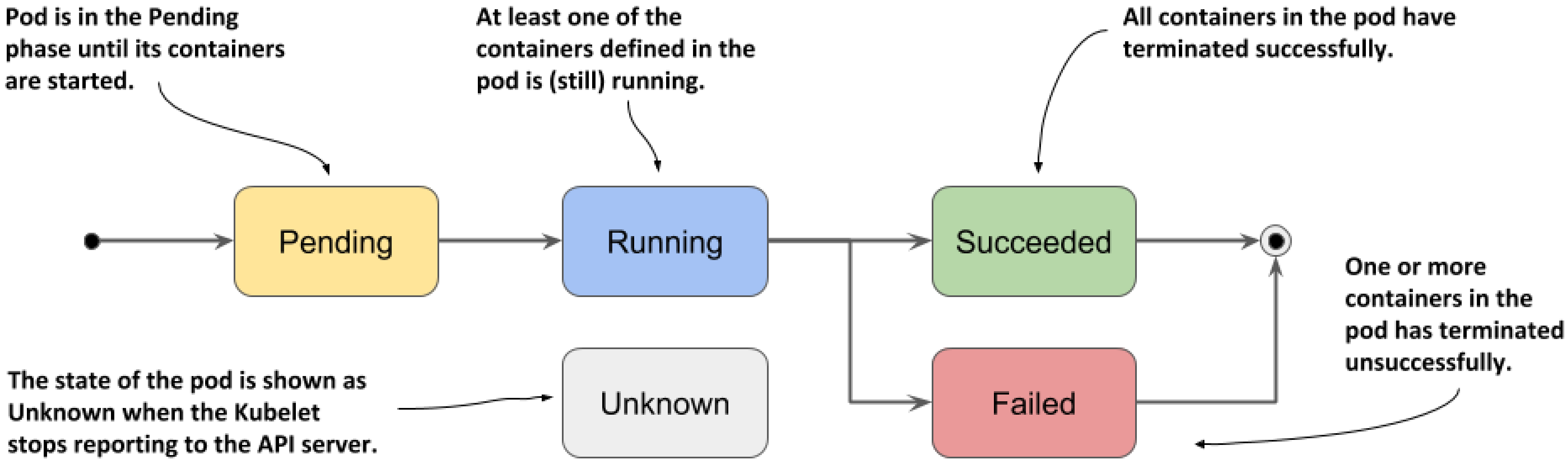
Workload observability (why?)

- Give developers or operations engineers live metrics for power usage
  - Optimize solutions for energy usage, E.g. comparing different approaches
  - Alerting after deploy → power usage went up 🚨
- CPU cycles is simple, kWh is approachable
  - Bottom-up: optimizing applications themselves
- First steps towards granular energy *efficiency*

Kubernetes Overview



Kubernetes Pod Lifecycle



Approach

1. Evaluate state-of-the art tool **Kepler** (CNCF Sandbox maturity)
   - Spoilers: not great…

Approach

1. Evaluate state-of-the art tool **Kepler** (CNCF Sandbox maturity)
2. Build and evaluate alternative **KubeWatt** based on *simple* power mapping model[1] as a proof-of-concept

[1] Andringa, L., Setz, B., & Andrikopoulos, V. (2025). Understanding the Energy Consumption of Cloud-native Software Systems. In Proceedings of the 16th ACM/SPEC International Conference on Performance Engineering (pp. 309–319). ICPE '25: 16th ACM/SPEC International Conference on Performance Engineering. ACM. https://doi.org/10.1145/3676151.3719371

Two “types” of power

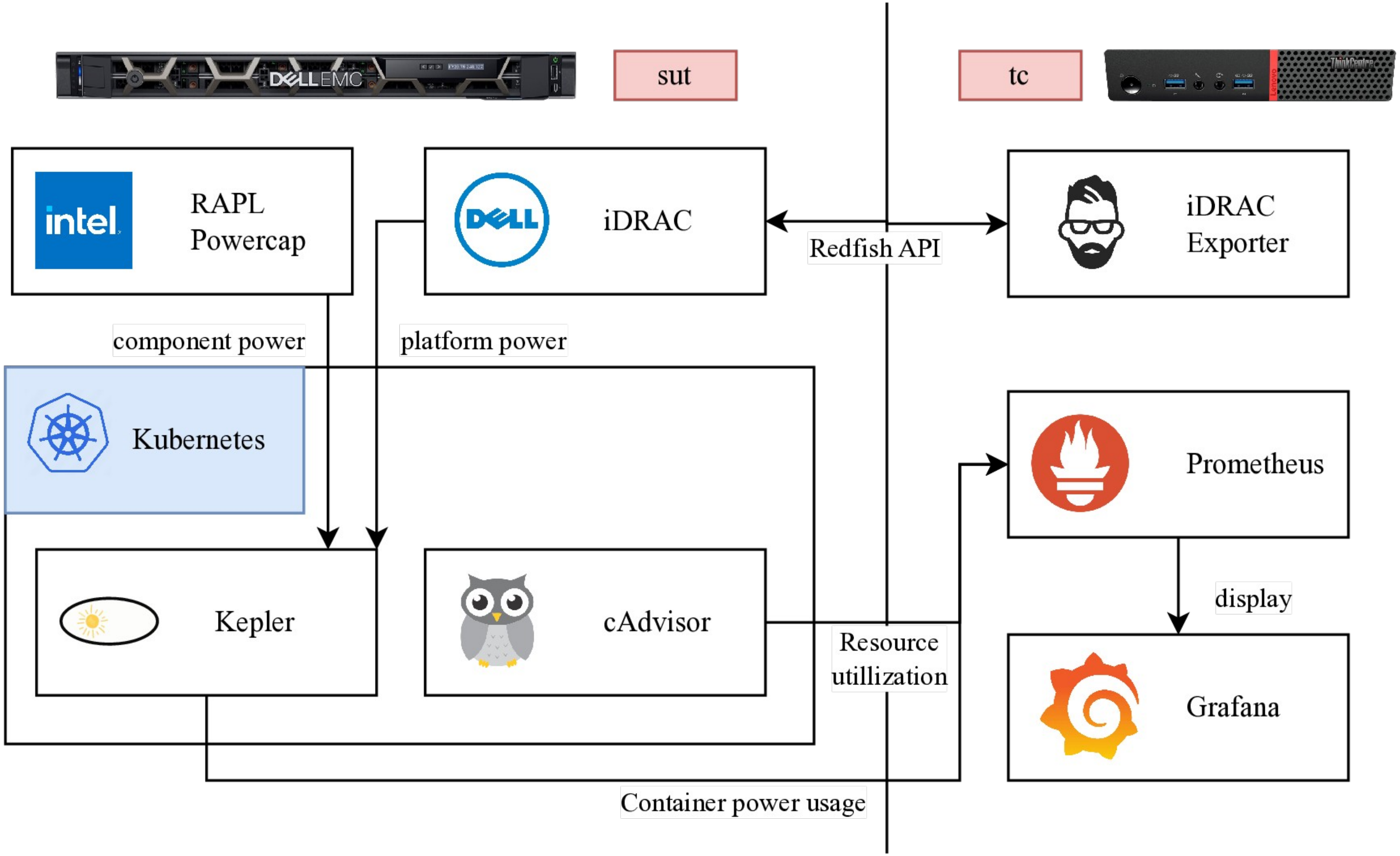
- System power *isn’t* a single number
- A system at rest uses power → **idle** (or static) power
  - Divided over all containers equally
- Resource utilization uses power → **dynamic** power
  - Attributed based on amount of resources used by container

CNCF Sandbox Kepler

- **K**ubernetes-based **E**fficient **P**ower **L**evel **E**xporter
- Estimates power usage per Kubernetes pod
- Measurements of platform and component power based on Intel RAPL, Redfish API and software estimation
  - Multiple or custom software estimation models available
- Export to Prometheus (/Grafana)

Experimental Validation

- Goal: Run Kepler with workloads and validate its estimations against real measurements
  - Accuracy of node metrics
  - Accuracy of container-level power metrics



Experimental Results

- Experimental validation shows significant drift from real-world measurements, e.g.:
  - Containers that were not running were attributed significant idle power usage → unrealistic
  - Power misattributed to non-existent processes, even though a single process is responsible for significant power usage → incorrect
  - Software estimation failed altogether
- Node metrics align precisely to the source data

Example > single stressor pod

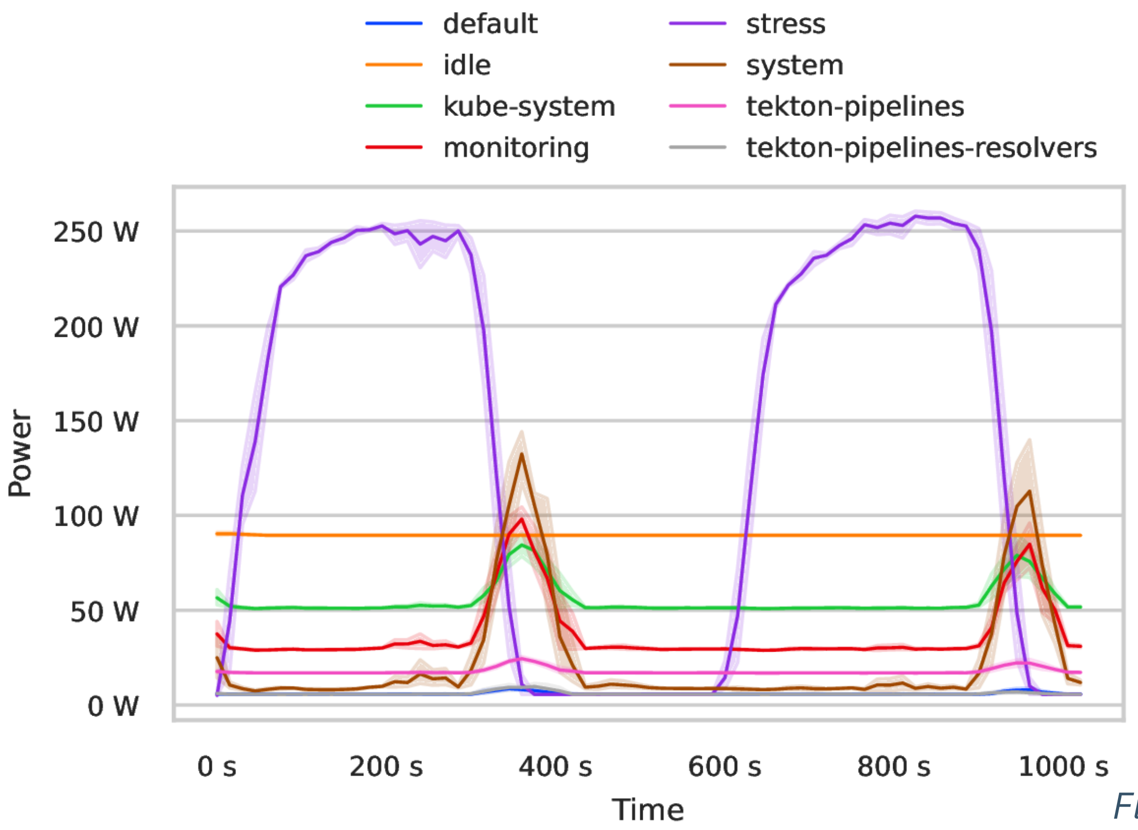


Figure 1

Example > ghost-containers using power

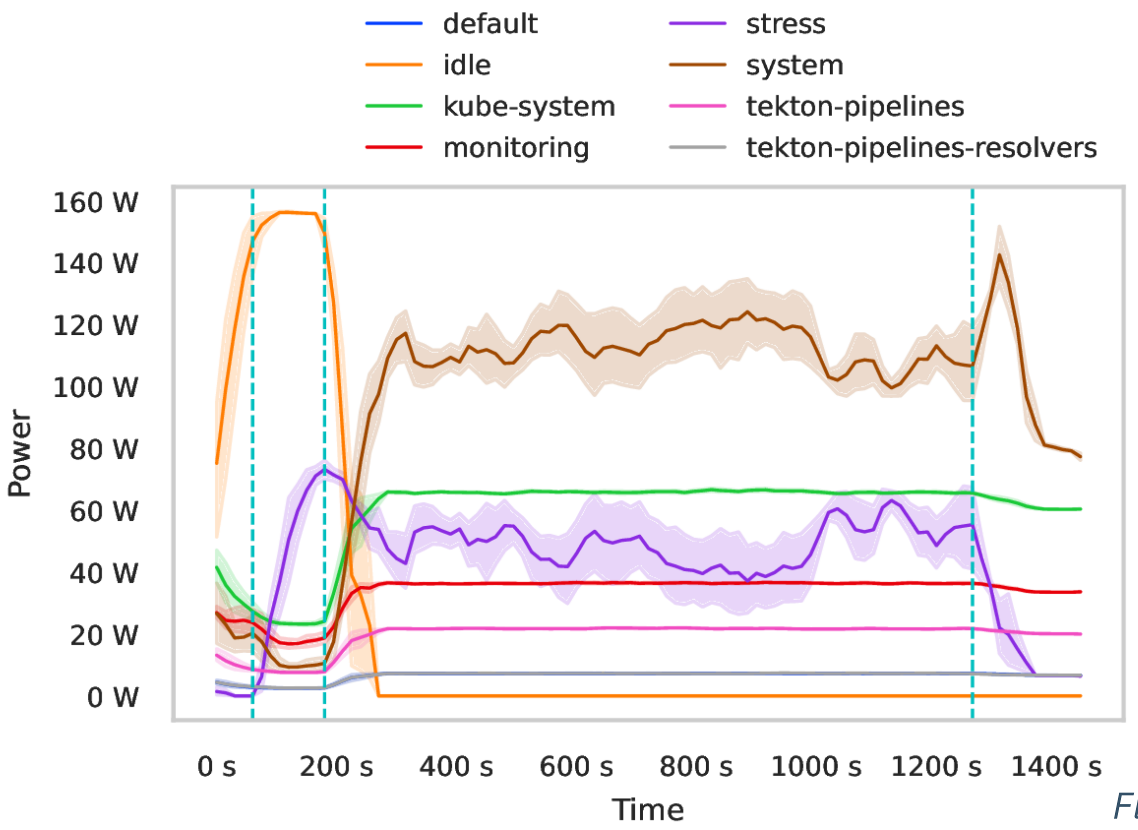


Figure 2

KubeWatt

- KubeWatt as a proof-of-concept
  - Validate that Kepler’s shortcomings are not inherent to the problem at hand
- Simple power mapping model:
  1) Determine dynamic power fraction
  2) Scale dynamic power value to CPU usage by container
- Data collected using **Redfish API** and **Kubernetes metrics** collector
- Export to Prometheus (/Grafana)

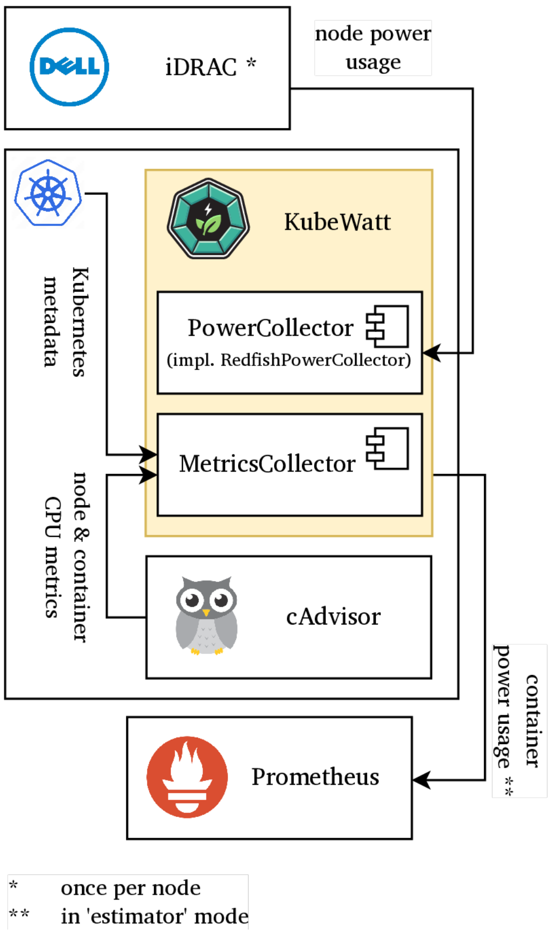
https://github.com/bjornpijnacker/kubewatt

Calculating static/dynamic power fraction

- Either:
  1) Measure the cluster at idle for some time
  2) Measure the cluster while running → find the value with regression
- Not an online measurement, cluster is expected not to change
  - Fine for short tests
  - Novel work on online power ratio model

KubeWatt estimation mode

- Divide dynamic power per node proportionally to container CPU utilization per node
- Due to its remarkably simple design, able to overcome the limitations shown for Kepler
  - Power attributed to the containers using the CPU
  - No power attributed to containers that are not running



Example, same test as Figure 2

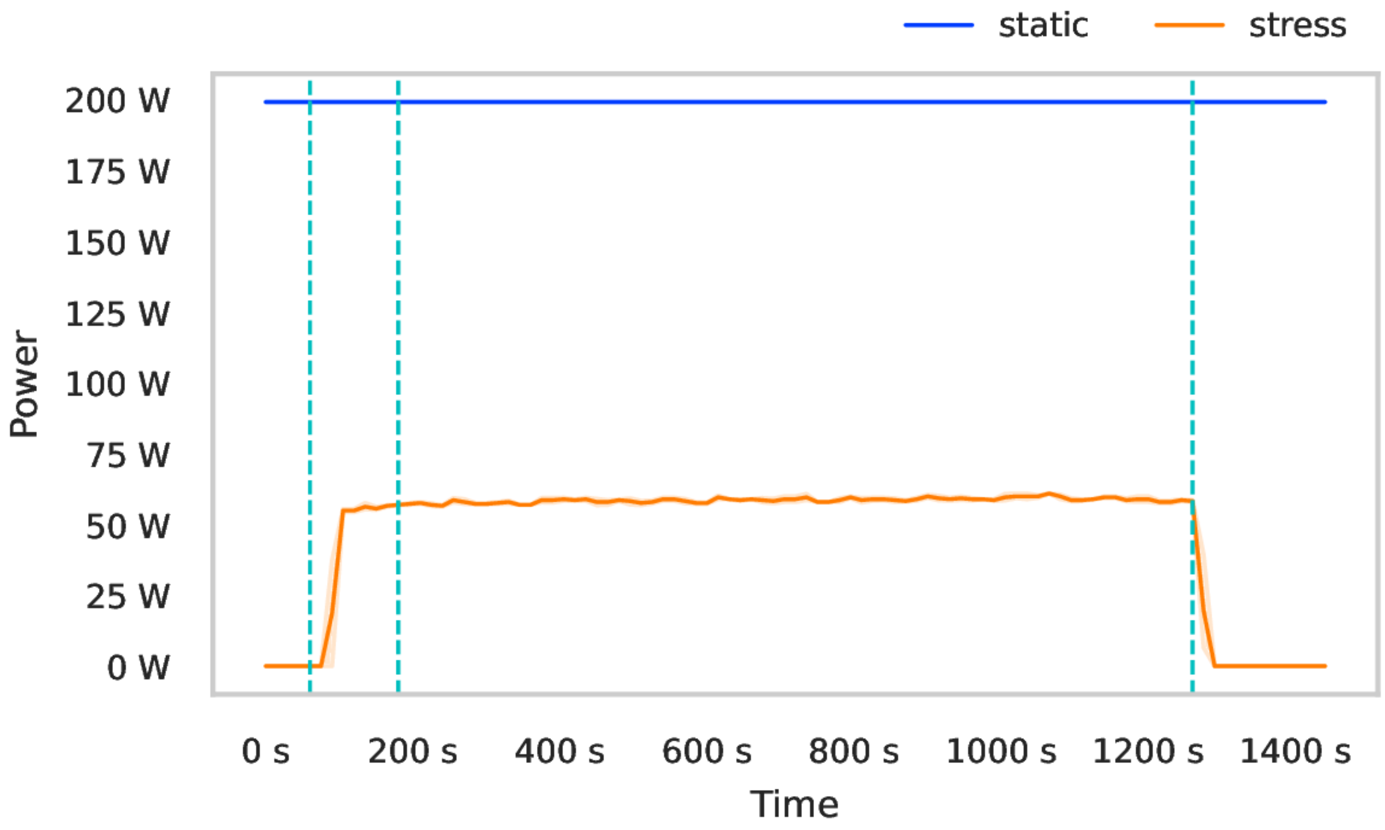


Figure 3

Estimation mode limitations

- Estimation based on real-world measurements
  - Not always available
  - No sufficient software solution exist (yet)
- Only supports CPU-based estimation
  - As a proof-of-concept
  - Largest power draw in non-GPU scenario’s

Container-Level Energy Observability in Kubernetes Clusters

ICT4S 2025
Dublin, Ireland

Bjorn Pijnacker[1]
Author & Speaker

Vasilios Andrikopoulos[1]
Co-author

Brian Setz[1]
Co-author

[1] University of Groningen