Containerization's Power Use Overhead in Video Streaming

Authors: E.V. Depasquale[1,2], S. Zammit[1]

[1] Department of Communications and Computer Engineering, University of Malta

[2] Contact email: edepa@ieee.org

L-Università ta' Malta

IARIA

Etienne-Victor Depasquale

Etienne-Victor Depasquale is an experienced network engineer and systems integrator with over twenty-five years in the field. In 2015, he transitioned into academia by joining the University of Malta, where he completed a PhD in 2024, focusing on standardized methods for reporting energy consumption in telecommunications networks.

His research interests lie at the intersection of sustainability, computing, and communications.

Aims and contributions of our paper

Aims

- Investigation of containerization's power overhead while streaming video
- Development of a re-usable method framework

Contributions

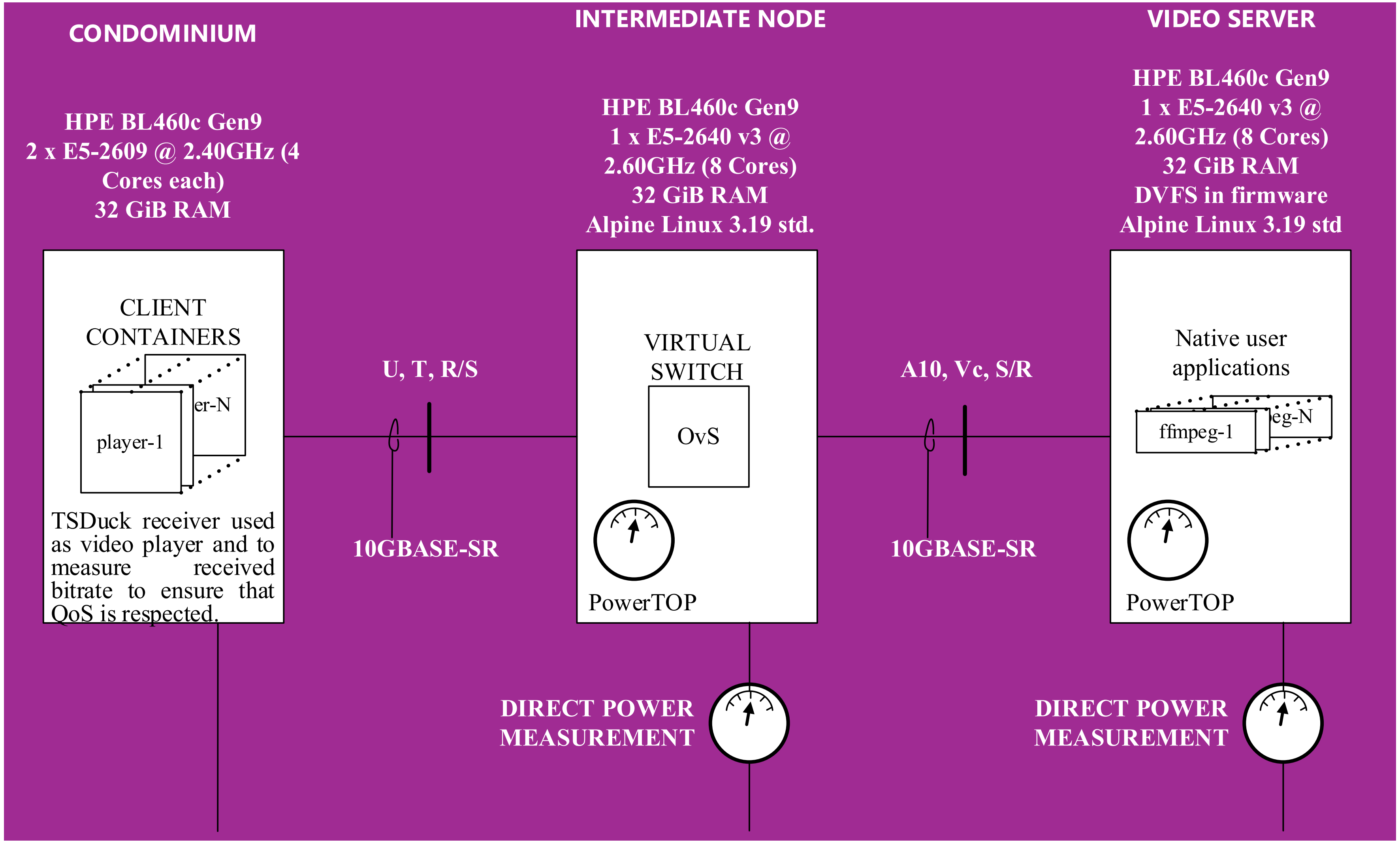
- Showed that containerized video service's power overhead is low.
- Demonstrated use and limitations of two existing power meters

Objective

- Quantify power use across two deployment scenarios:
  - Containerized video streaming server.
  - Native OS-based video streaming server.
- Trade-off: Operating power per unit (host) vs. potential for consolidating services on fewer physical hosts.

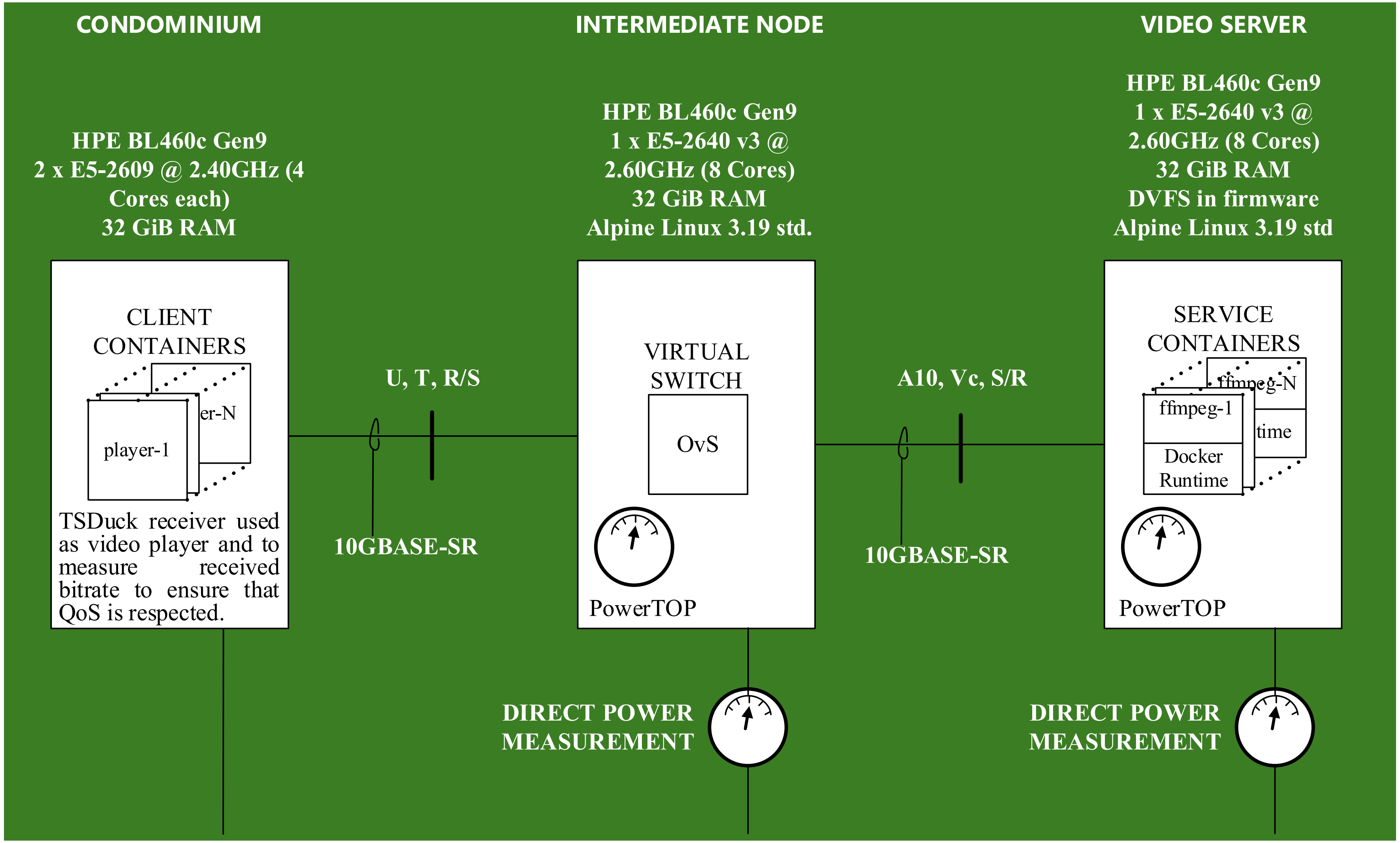
Implementation Model: Classical form

- Video server runs in access node (AN) as application on host OS
- Active Ethernet access network
- Described using standardized reference points



Implementation Model: Cloud-native form

- Video server runs in access node (AN), containerized
- Active Ethernet access network
- Described using standardized reference points



Method: Instrumentation

- Two power meters, for cross-validation
- Integrated Lights-Out (iLO) Redfish-compliant RESTful API, with 10-second averaging period
  - Excludes measurement of fan and ac-dc overhead
- PowerTOP
  - Dynamic power measurement only
  - Hard disk drive and solid-state disk power use is not captured
  - Mitigate by logging to RAM disk and by capturing static power baseline

Method: Baseline capture

- Denote dynamic power corresponding to OS operation with container system software running by $P_q^{(os+dockerd+containerd)}$
- Denote idle/leakage/static power at the frequency f_2 at which the OS is quiescent by $P_{idle}^{f_2}$
- Desired baseline is $P_{b_2}^{(video)} = P_{idle}^{f_2} + P_q^{(os+dockerd+containerd)}$
- Add this to PowerTOP and compare with iLO's measurement

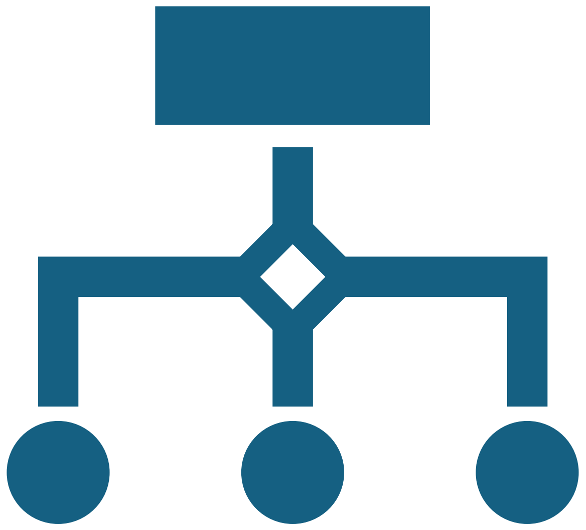
Method: Quality of Service (QoS)

- Measure received rate with fine granularity (5 s)
- Compare average of fine-grained rate samples with overall average bit rate for video and audio.
- TSDuck used for this purpose.

Experimental procedure: Orchestration

Automated management using Python scripts and Ansible

1. Reboot video server, to obtain common and reproducible initial state.
2. Wait until the video server quiesces.
3. Start the power meters for both total and dynamic power, for both the video server and the virtual switch.
4. Wait for a fifteen-minute interval, to capture behaviour before video streaming.
5. Instantiate & start an Ffmpeg listener container[1], poised for real-time play-back with randomized starting point & 30-minute play time.
6. Start a TSDuck client to connect to the container and measure the bitrate, averaged over 5-second intervals.
7. Once 30 minutes of video have been played, destroy the container.
8. Wait for a fifteen-minute interval, to capture behaviour after video streaming.



Note 1. In the classical form, start the listener as a new process native to the host OS.

Experimental procedure: Concurrency

For several concurrent streams, steps 6 and 7 must be repeated for each one of the additional streams.

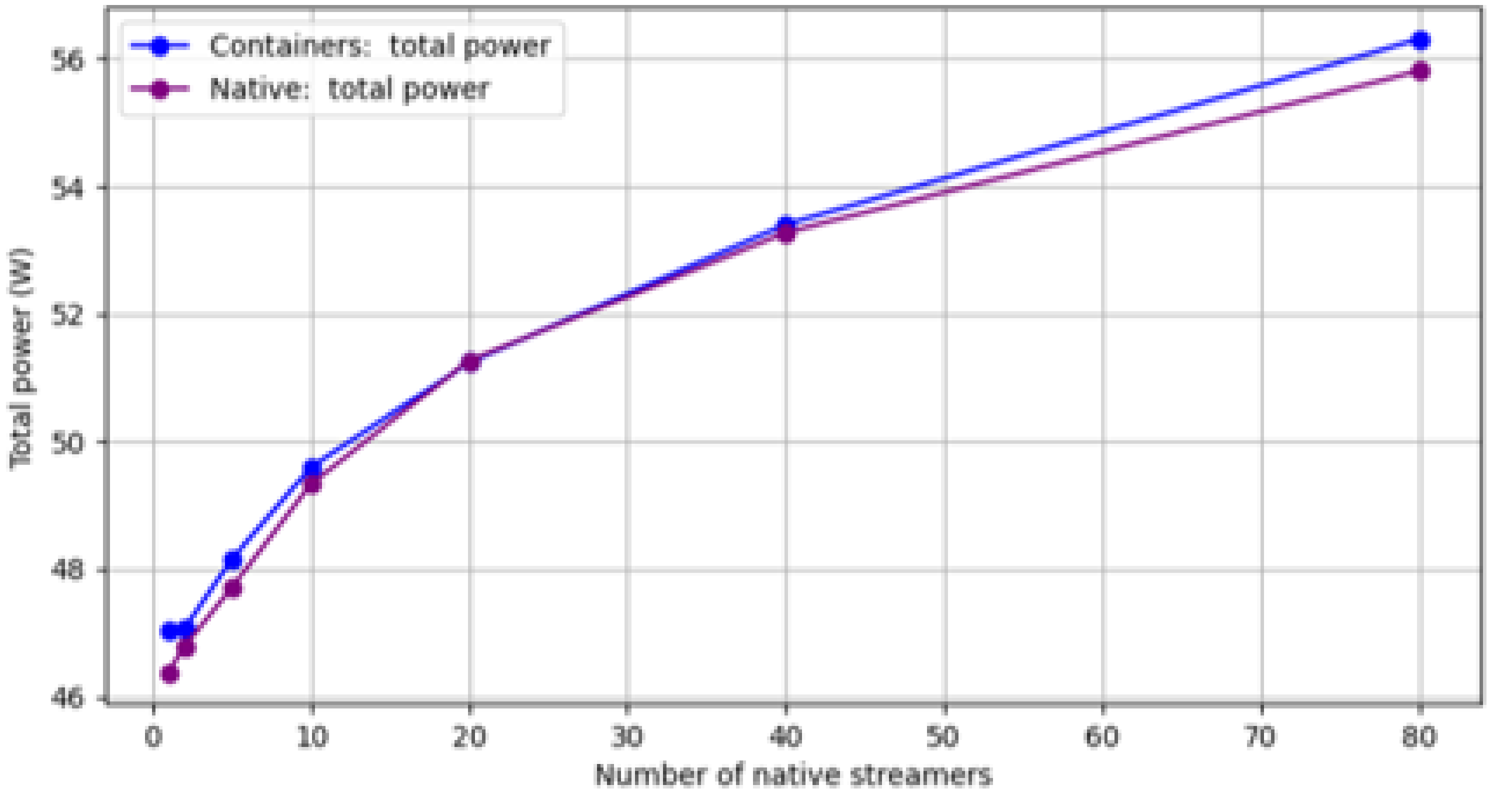
For the native service instance, step 5 involves the ffmpeg process only and there is no equivalent to step 7.

Number of concurrent instances varied between 1 and 80: 1, 2, 5, 10, 20, 40, 80.

Results – Power Comparison

Minimal total power overhead in containerized setups

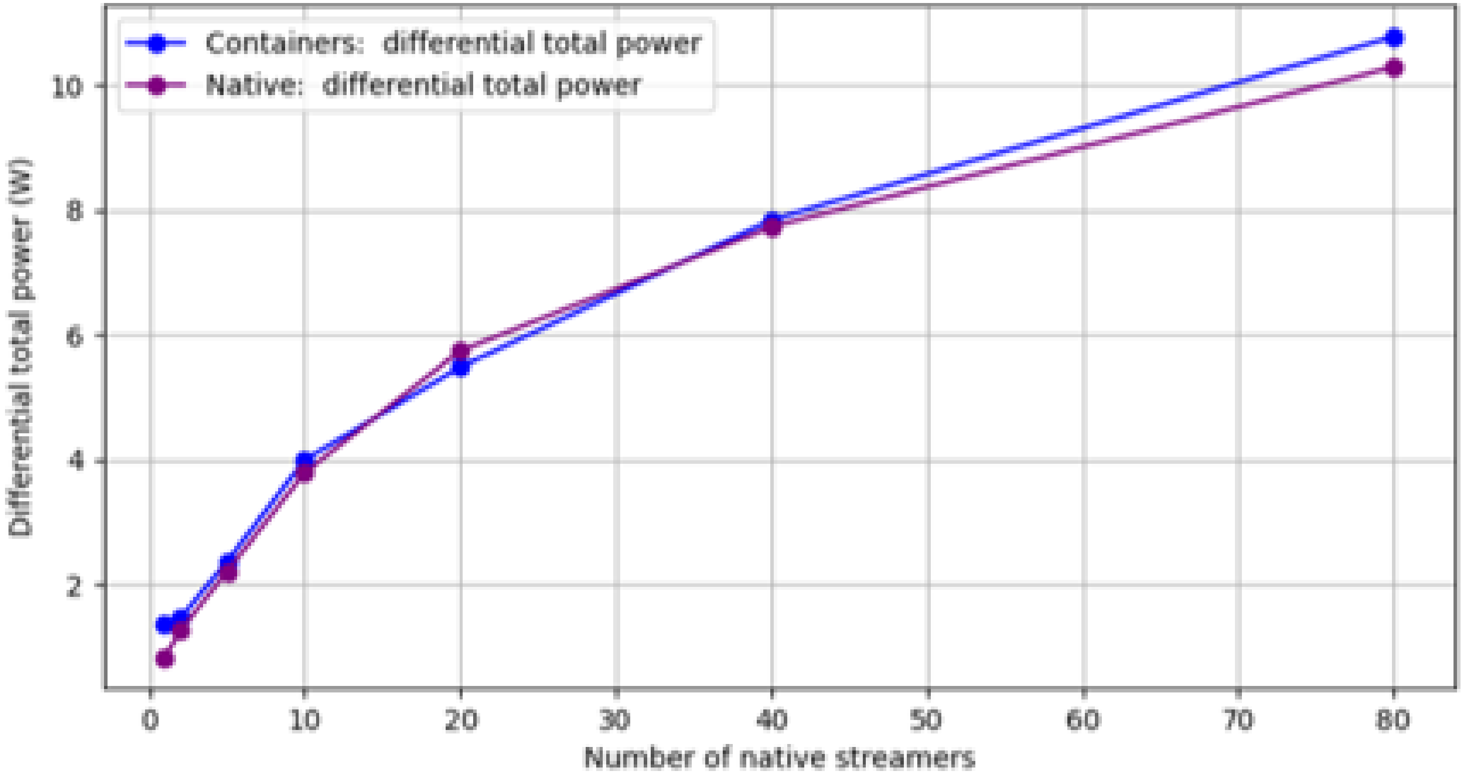
Total power during operations measured at iLO, $P_{ops}^{iLO}(n)$ vs **number of streamers, n**



Differential (dynamic) power

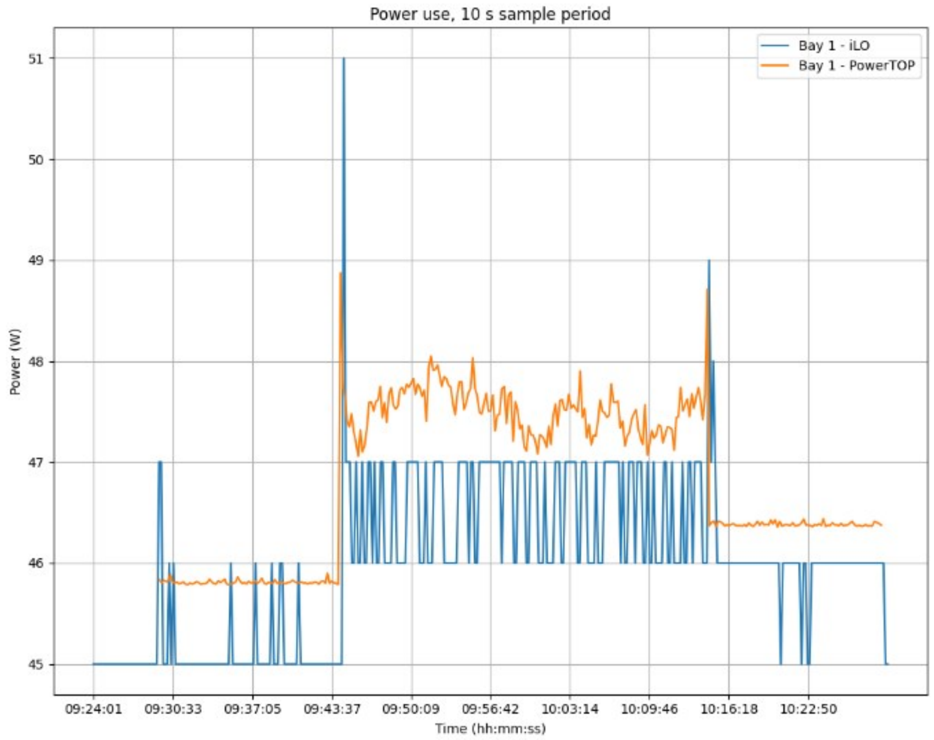
- Differential power: between operations and quiescence, $P_{ops}^{iLO}(n) - P_q^{iLO}$.
- Minimal overhead in differential power

Differential power during operations measured at iLO
$\boldsymbol{P_{ops}^{iLO}(n) - P_q^{iLO}}$,
vs **number of streamers, $\boldsymbol{n}$**

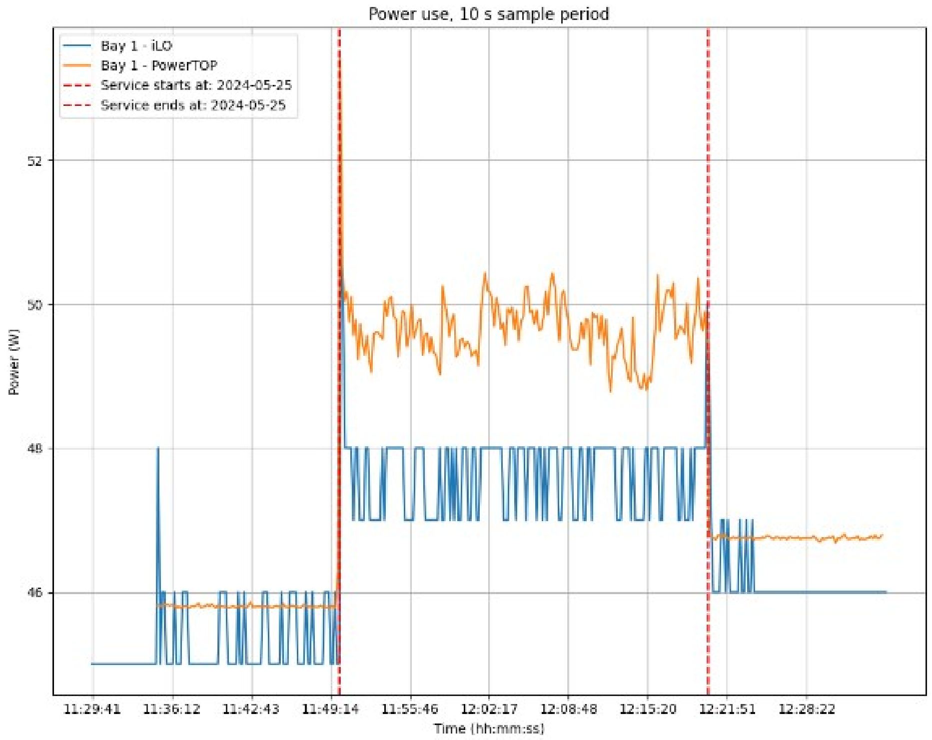


PowerTOP's accuracy (1)

Absolute error grows with average power use

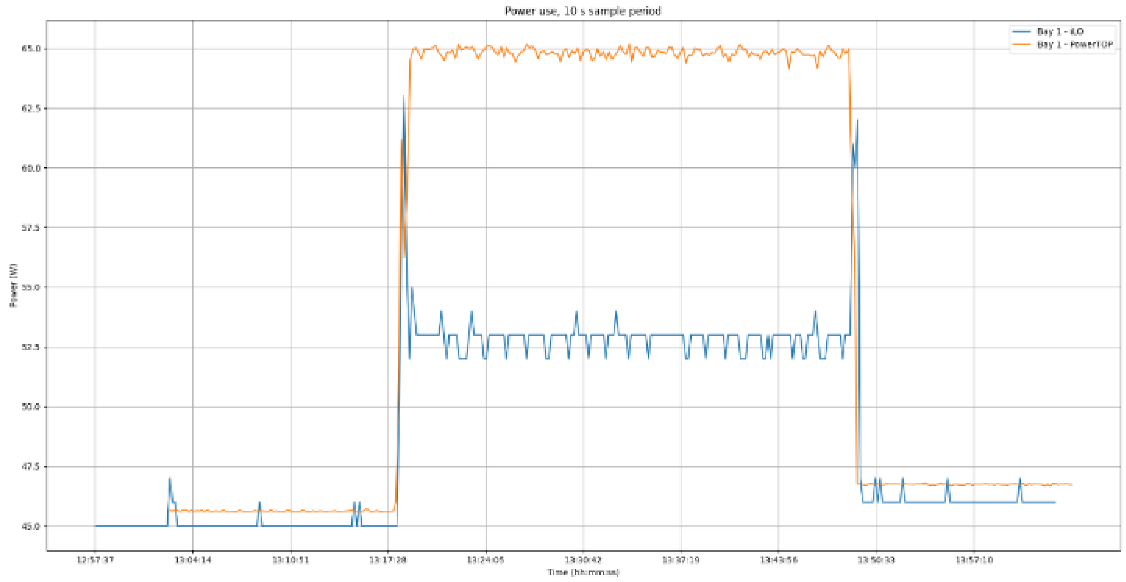


2 instances

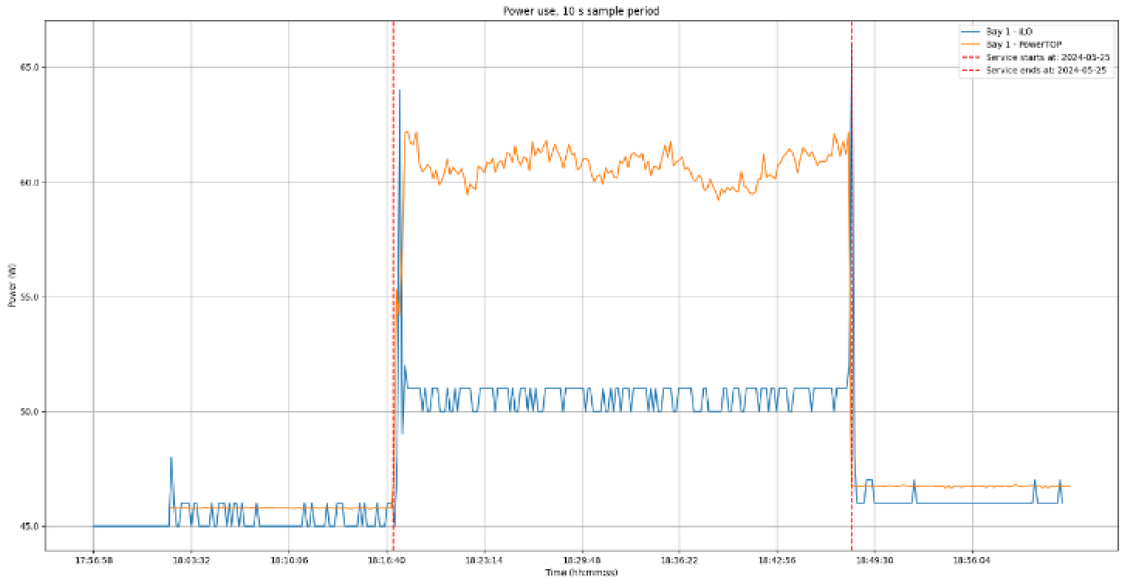


5 instances

PowerTOP's accuracy (2)



20 instances



40 instances

Conclusion

Containerization poses minimal overhead for video edge caches

Energy-efficient for scalable deployment.

Scope for future work: Calibration of PowerTOP for greater accuracy.