2025 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE

MUNICH, GERMANY
OCTOBER 14-15, 2025

Performance Analysis of Federated Simulations using the Open-Source SIL Kit Library

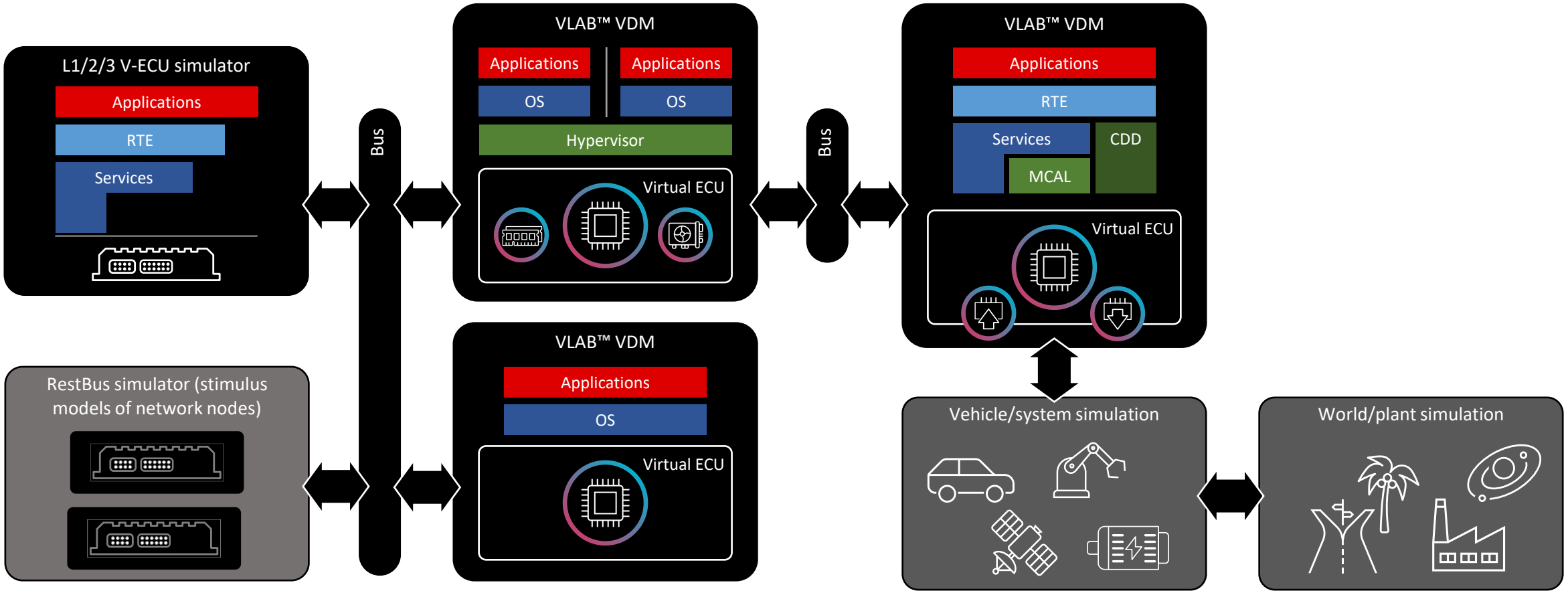
Jakob Engblom, Cadence (VLAB Works)

engblom@cadence.com

VLAB WORKS™ cadence®

accellera SYSTEMS INITIATIVE

The Goal: Federated Simulation



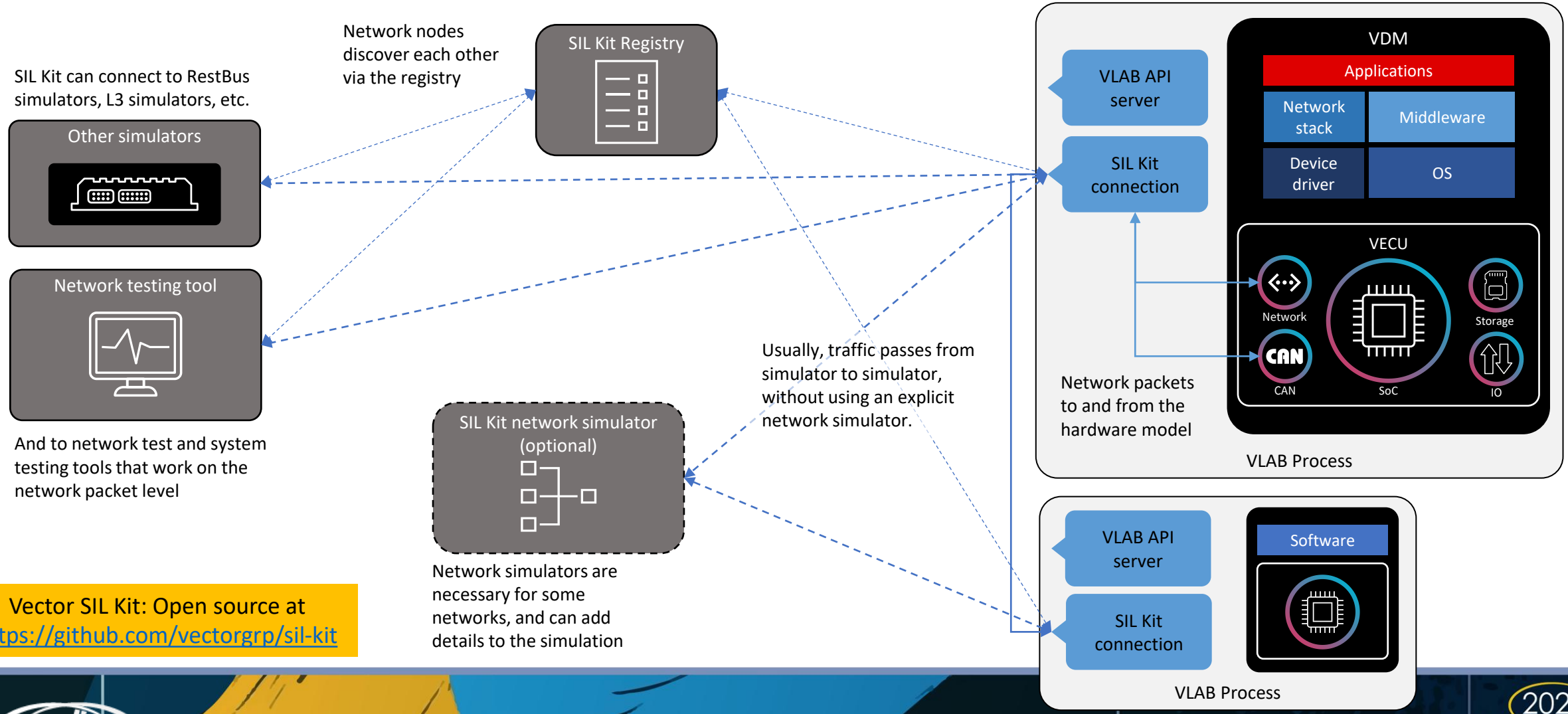
2025 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE

MUNICH, GERMANY
OCTOBER 14-15, 2025

Networking with VLAB

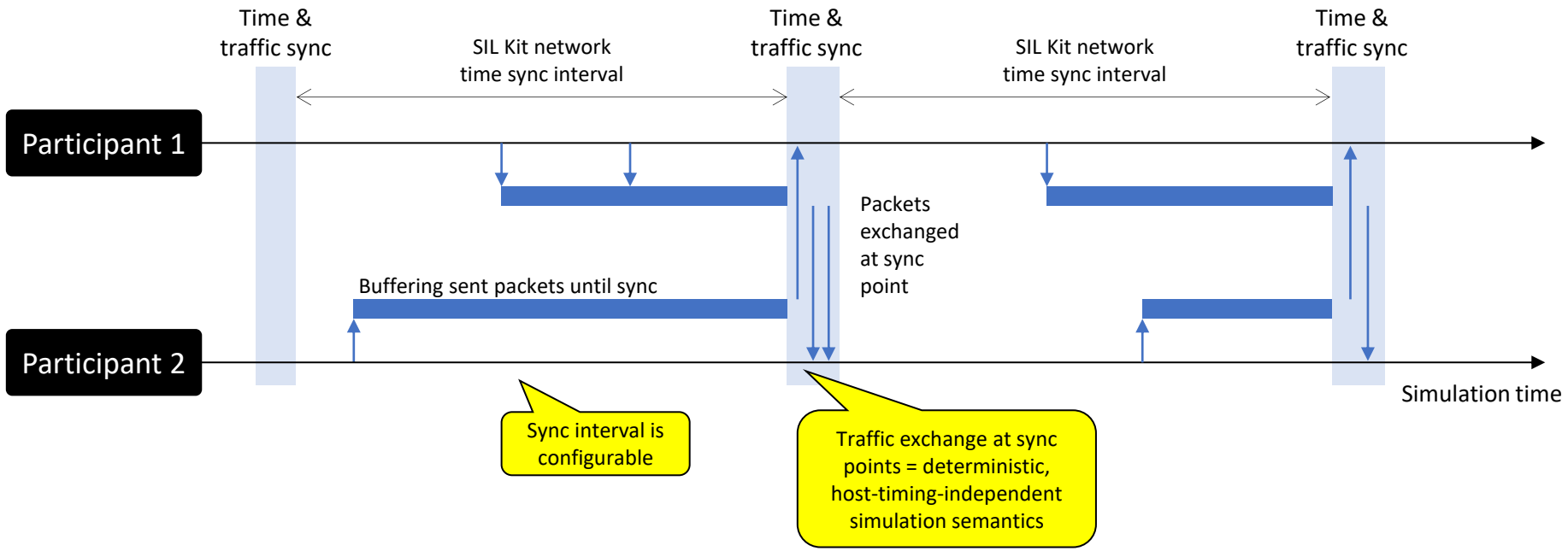
accellera SYSTEMS INITIATIVE

Networking VLAB with SIL Kit



Vector SIL Kit: Open source at https://github.com/vectorgrp/sil-kit

SIL Kit Semantics



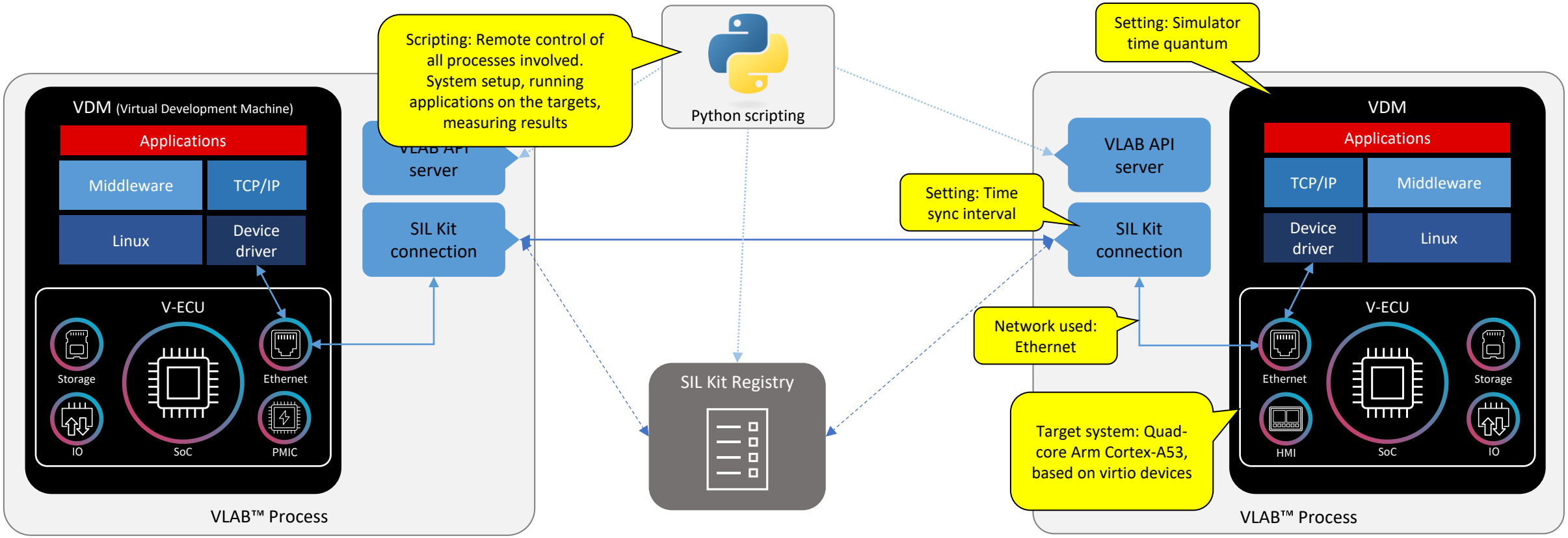
2025 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE

MUNICH, GERMANY
OCTOBER 14-15, 2025

Experimental Setup

accellera SYSTEMS INITIATIVE

Experiment, for Two VDMs



Scripting

```
# SIL Kit time intervals
siltimes = [vlab.us(2), … ]

# Time quanta
quatimes = [vlab.us(1), …]

num_machines = … # from shell
repeats = …  # 2 to 4, depending

for silkit_quantum in siltimes:
    for time_quantum in quatimes:
        for r in range(repeats):
            run_experiment(…)
```

1 us = 2000 processor cycles, time quanta are long enough to not interfere with core ISS performance

```
def run_experiment(num_machines, time_quantum, silkit_quantum, …):
    start_silkit_registry()
    start_silkit_controller()  # tracks machines in simulation
    for i in range(num_machines):
        start_machine(i, time_quantum, silkit_quantum)
    # Experiment body
    # - Script target serial input on all machines, etc.
    # - Wait for run to complete
    # - Save measurements

    # Clean-up: stop (exit) all processes started
    for i in range(num_machines):
        stop_machine(i)
    stop_silkit_controller()
    stop_silkit_registry()
```

2025 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE

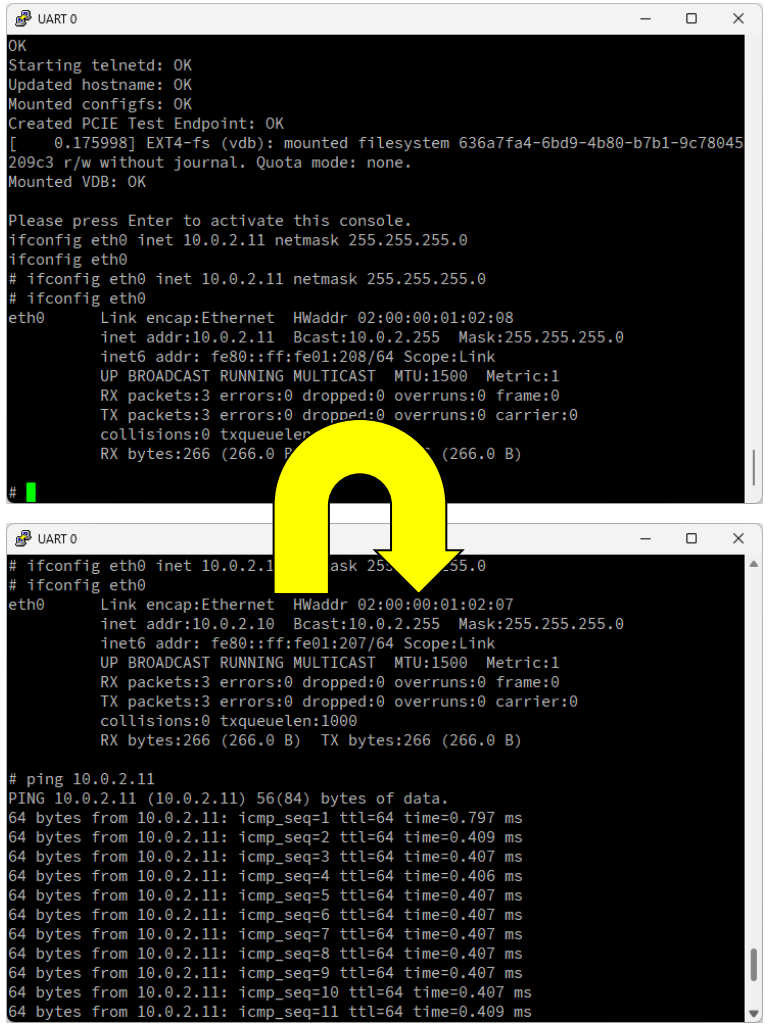
MUNICH, GERMANY
OCTOBER 14-15, 2025

Results

accellera SYSTEMS INITIATIVE

Basic Test: Ping

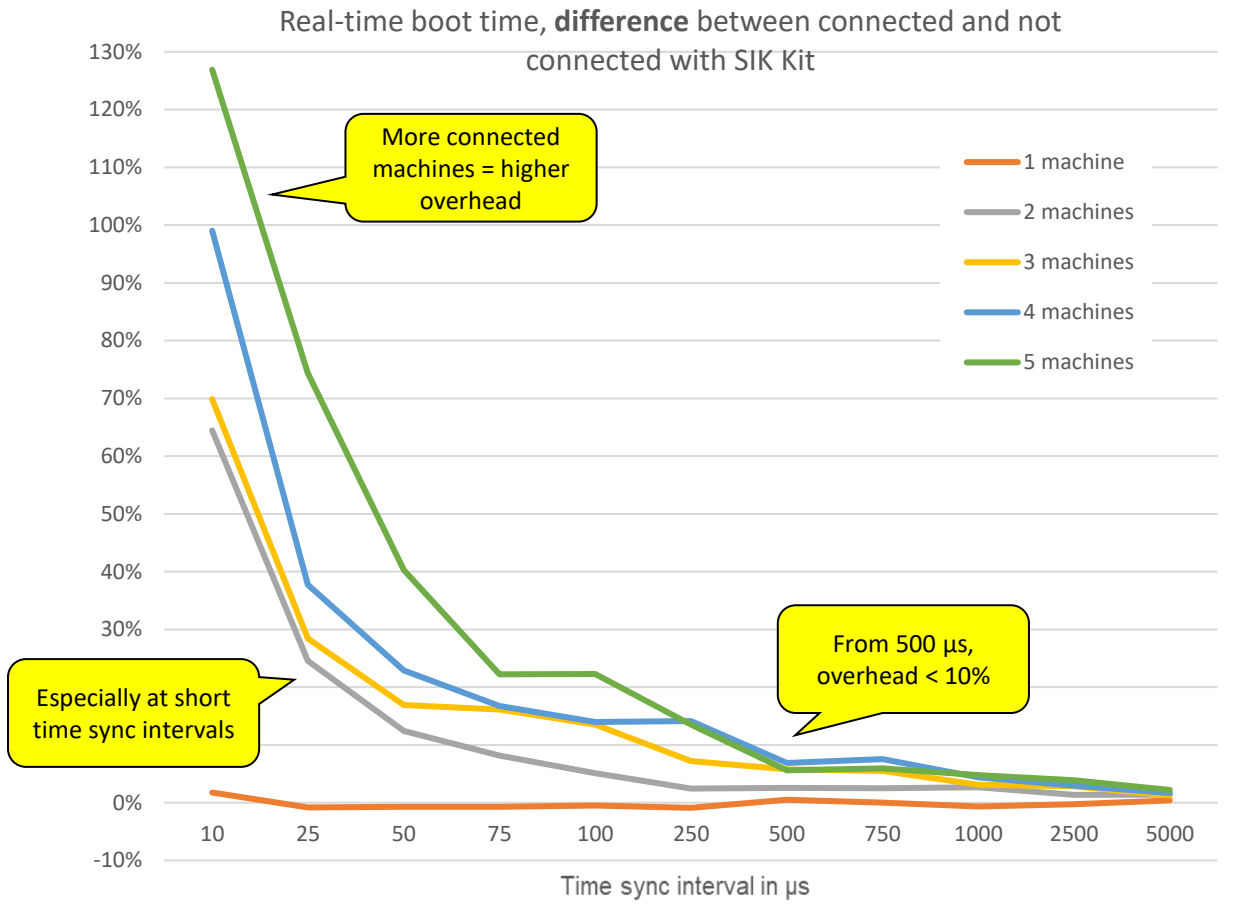
- Ping from one machine to the other
- Check the reported round-trip time
  - While varying the SIL Kit Time Sync Interval
- Results:
  - Longer sync interval = longer ping time
  - Ping time proportional to sync interval
- Observations:
  - The network latency is visible to software (duh)
  - Ping latency independent of the host speed (just checking)



SIL Kit latency is visible to software

SIL Kit Sync Overhead

- Assess overhead caused by the regular time sync between the simulators
- Booting N VDMs
  - In N processes
  - Connected or not connected
- Measure:
  - Compare the time to complete the boot when connected and when not connected



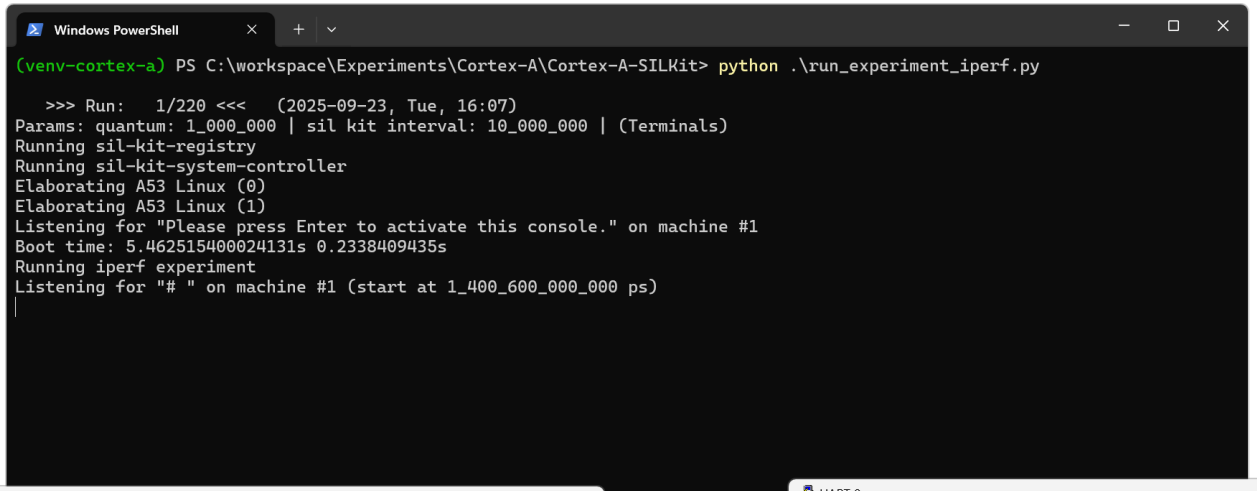
Overhead increases with more machines and shorter sync times

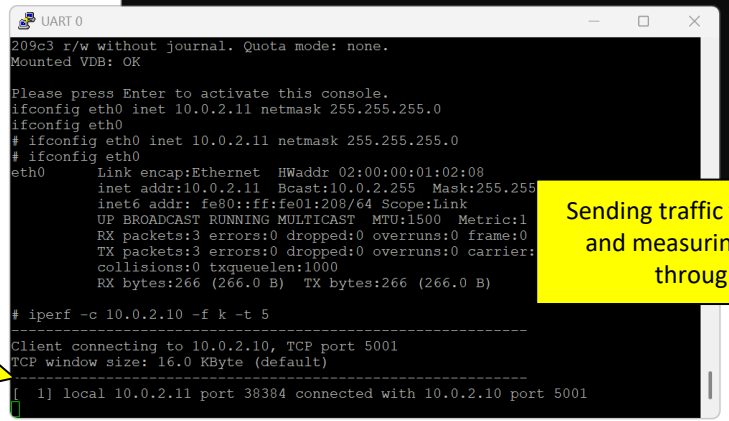
Main Experimental Workload: iPerf2

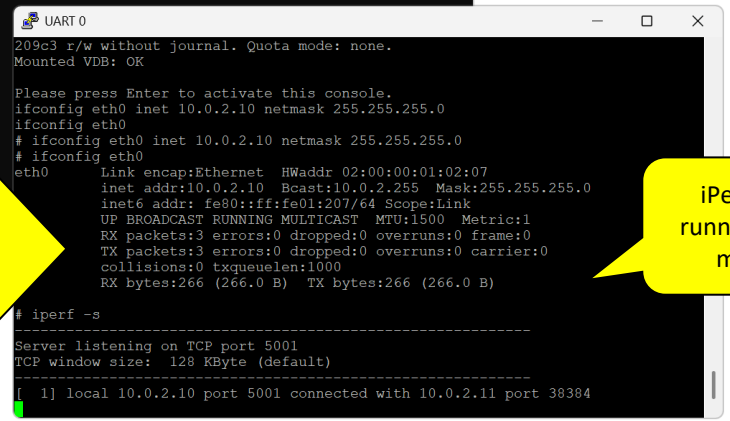
Measure the time to boot the target system – virtual and wall-clock **time**

Measure the time of the iperf run – virtual and wall-clock **time**

Record the **throughput** reported by iperf







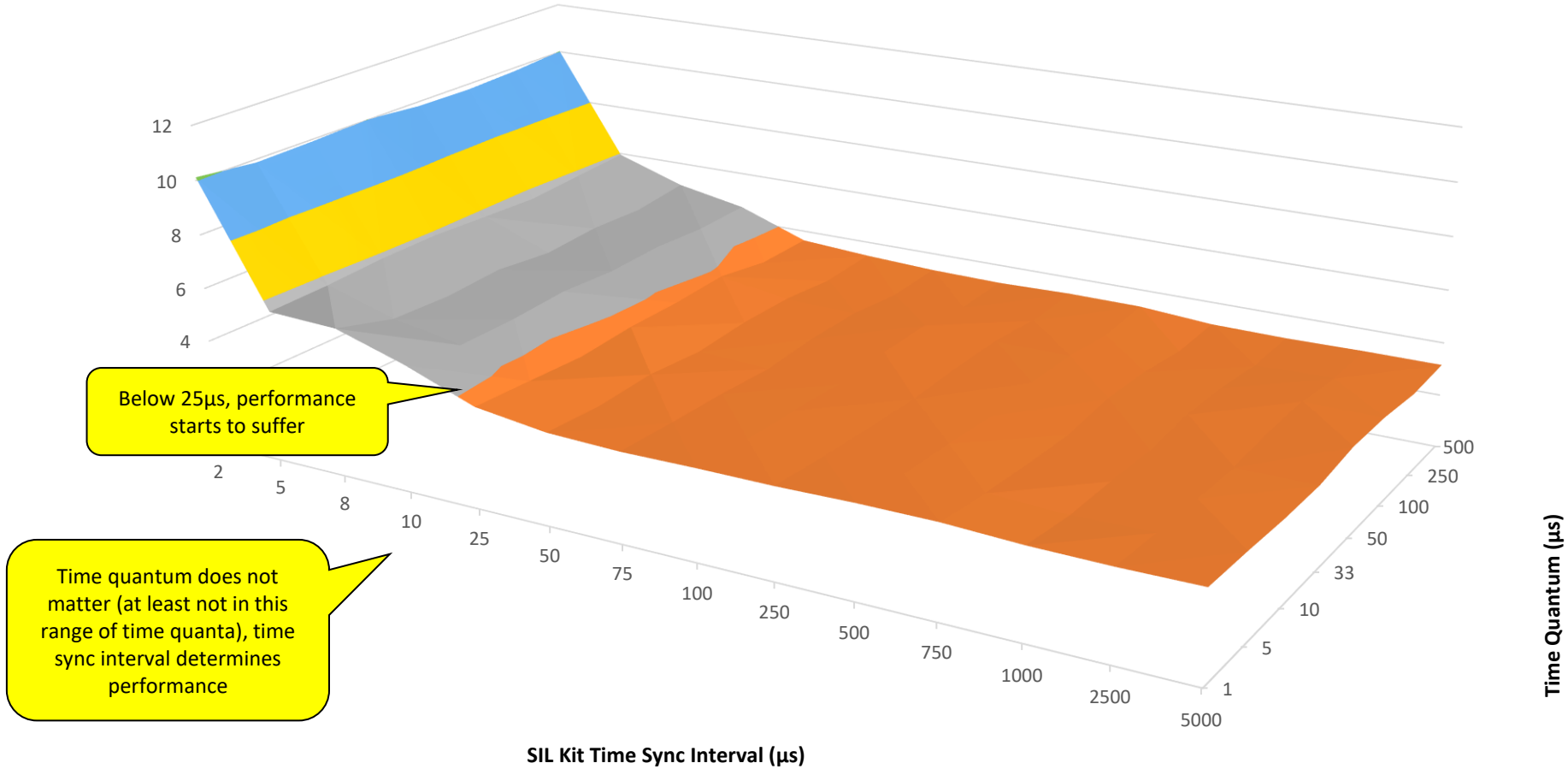
Default iperf parameters (TCP), running for 5 virtual seconds

iPerf client running on the other machine

Sending traffic to the server and measuring achieved throughput

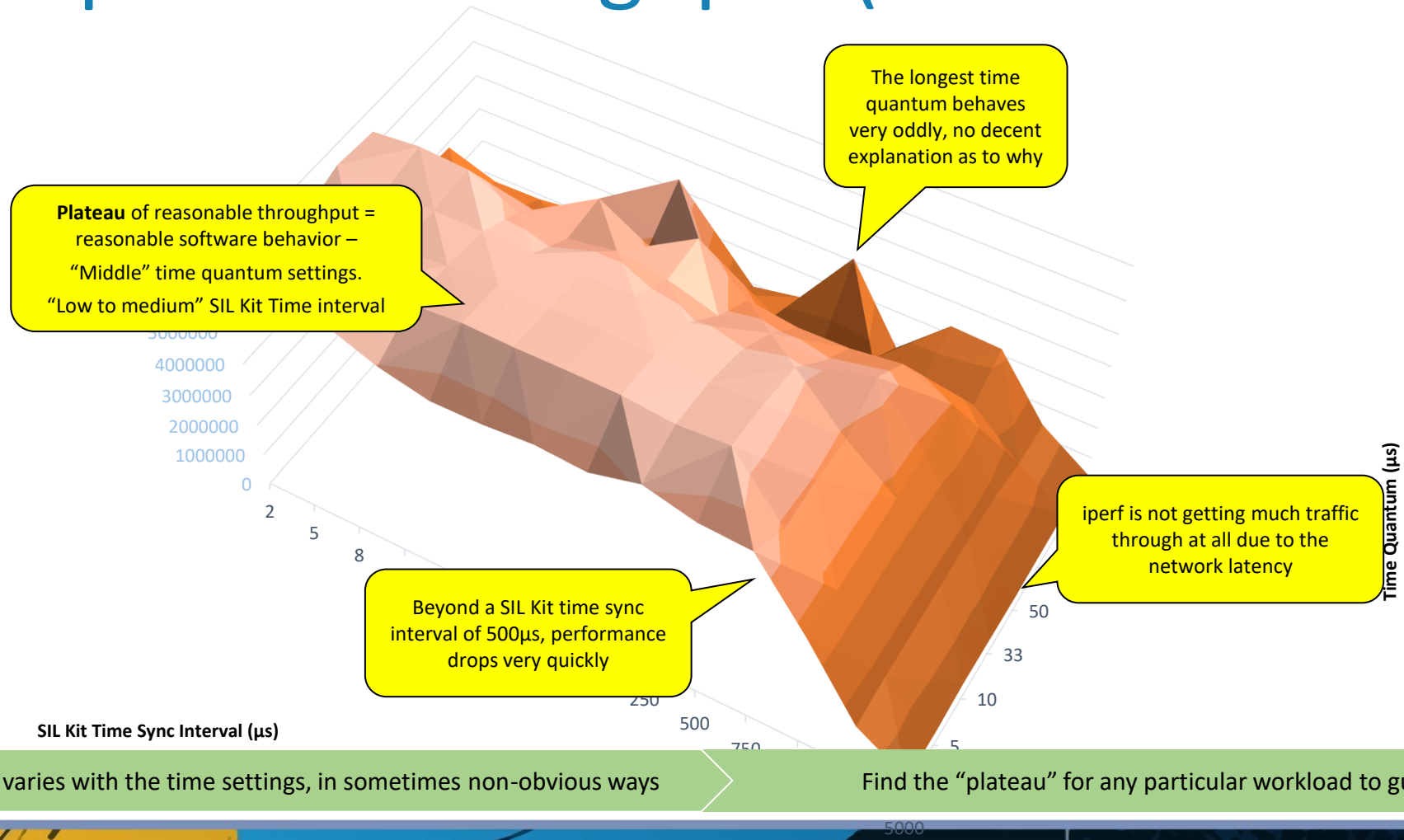
iPerf server running on one machine

Linux Boot Run Time



With no network traffic, optimal SIL Kit time sync interval is “make it larger”… but then why sync at all?

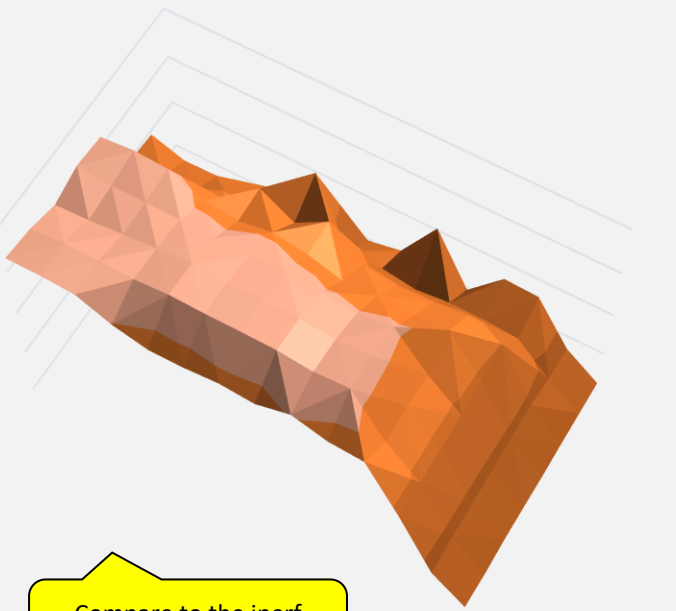
iPerf-Reported Throughput (“Performance”)



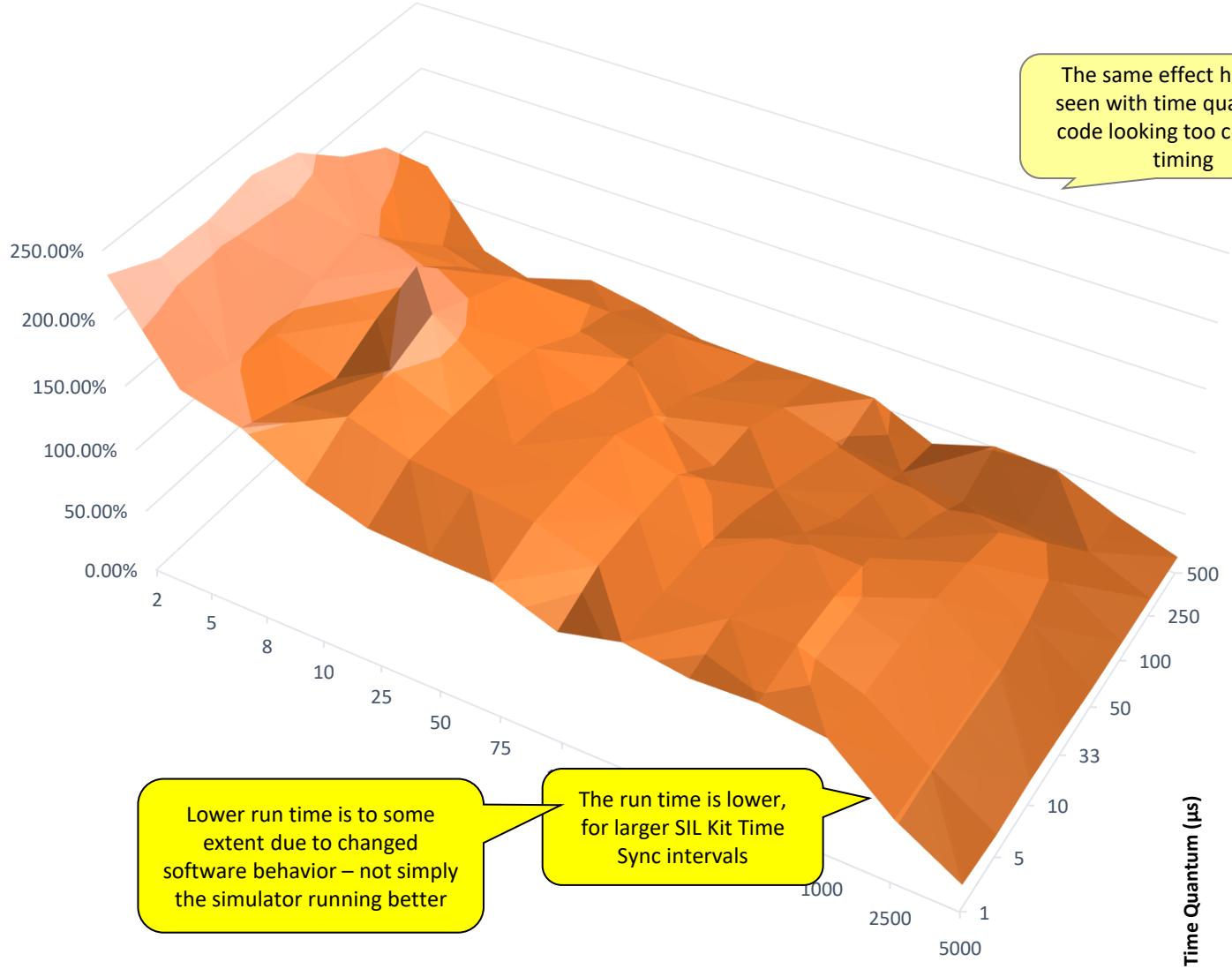
The software-visible behavior varies with the time settings, in sometimes non-obvious ways

Find the “plateau” for any particular workload to guide settings

iPerf Run Time



Compare to the iperf throughput profile



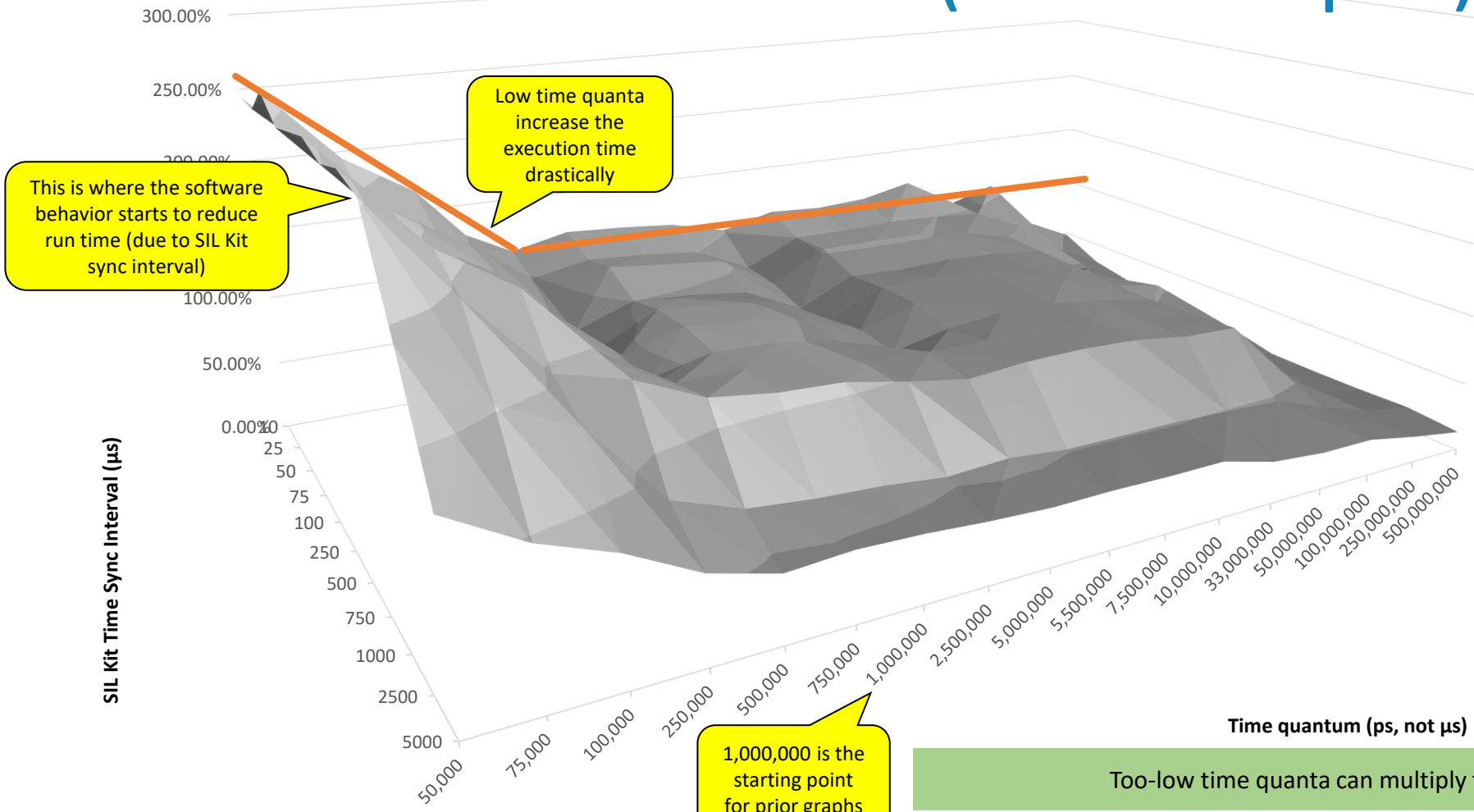
The same effect has been seen with time quanta and code looking too closely at timing

Lower run time is to some extent due to changed software behavior – not simply the simulator running better

The run time is lower, for larger SIL Kit Time Sync intervals

Lower run time is only meaningful if the software behavior remains the same

Extended iPerf Run Time (Not in Paper)



Using a different version of the platform = slightly different bumps in the landscape

Too-low time quanta can multiply the execution time

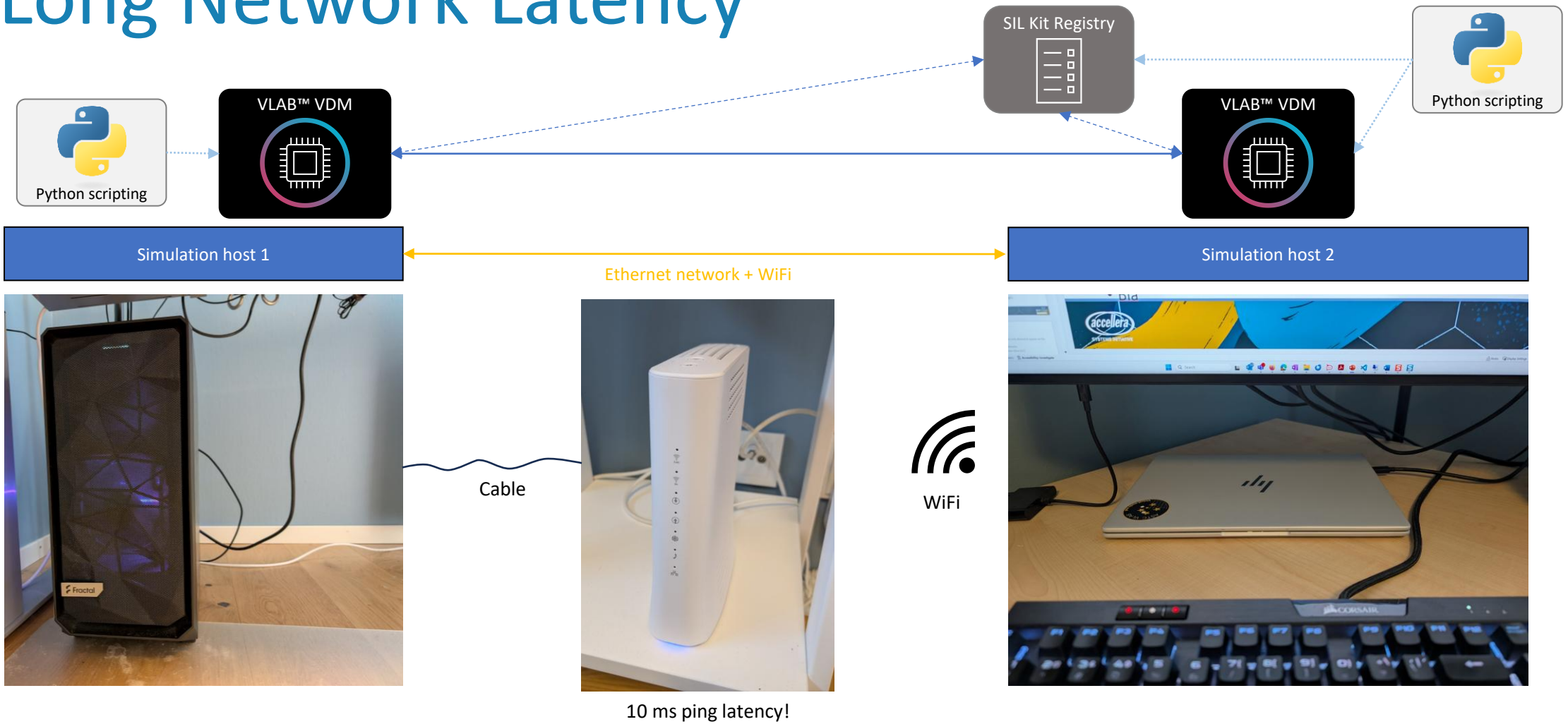
2025 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE

MUNICH, GERMANY
OCTOBER 14-15, 2025

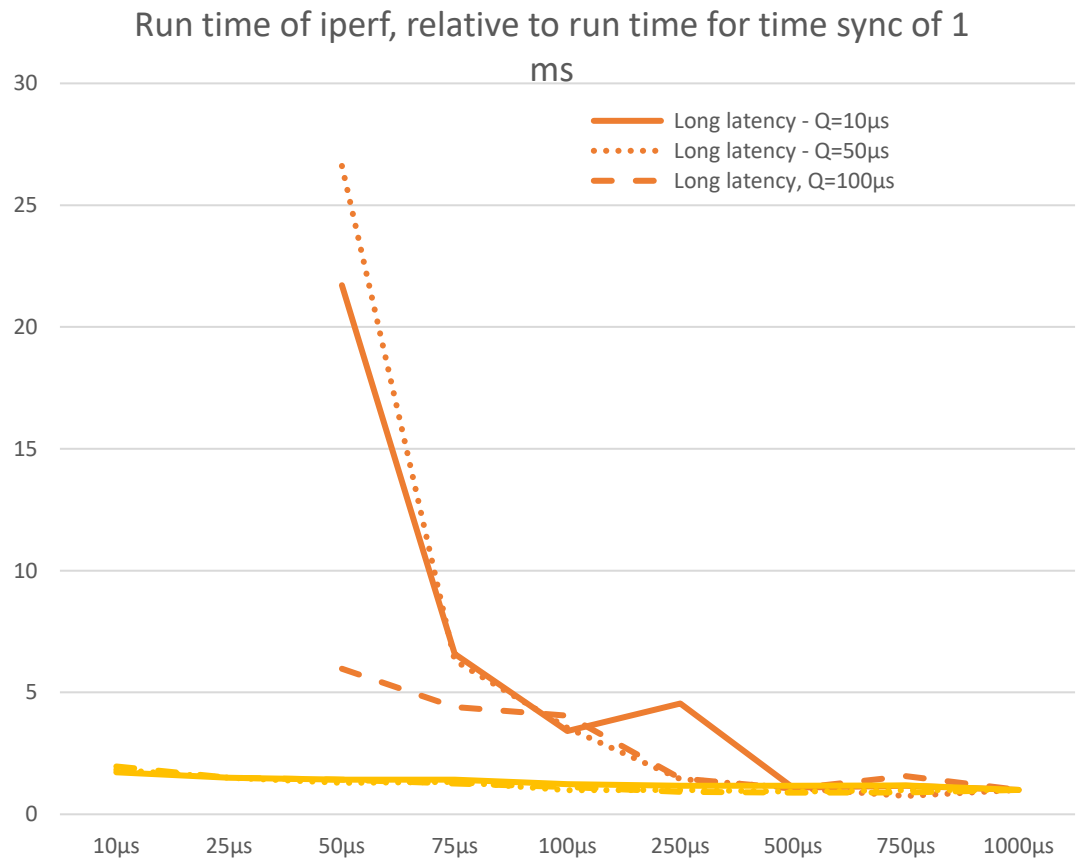
Distributed Simulation

accellera SYSTEMS INITIATIVE

Long Network Latency



Results



- This got bad quickly
  - Longest runs took 1 hour of wall-clock time!
    - Did not try below 50 µs
- Processes essentially spend all their time idling/waiting

High host network latency can kill performance

2025 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE

MUNICH, GERMANY
OCTOBER 14-15, 2025

Conclusions

accellera SYSTEMS INITIATIVE

Conclusions and Advice

Network latency affects software behavior

Longer network time sync intervals reduce overhead *(especially when running many machines)*

Time quanta less important than network time sync interval *(as long as the value is reasonably high)*

Understand the software behavior landscape – measure across a range of time settings

Ensure shortest-possible host-network latency between processes

Strive for longest possible network time sync intervals *(depends on application)*

2025 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE

MUNICH, GERMANY
OCTOBER 14-15, 2025

Questions?

accellera SYSTEMS INITIATIVE

2025 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE

MUNICH, GERMANY
OCTOBER 14-15, 2025

Backups

accellera SYSTEMS INITIATIVE

iPerf Throughput