



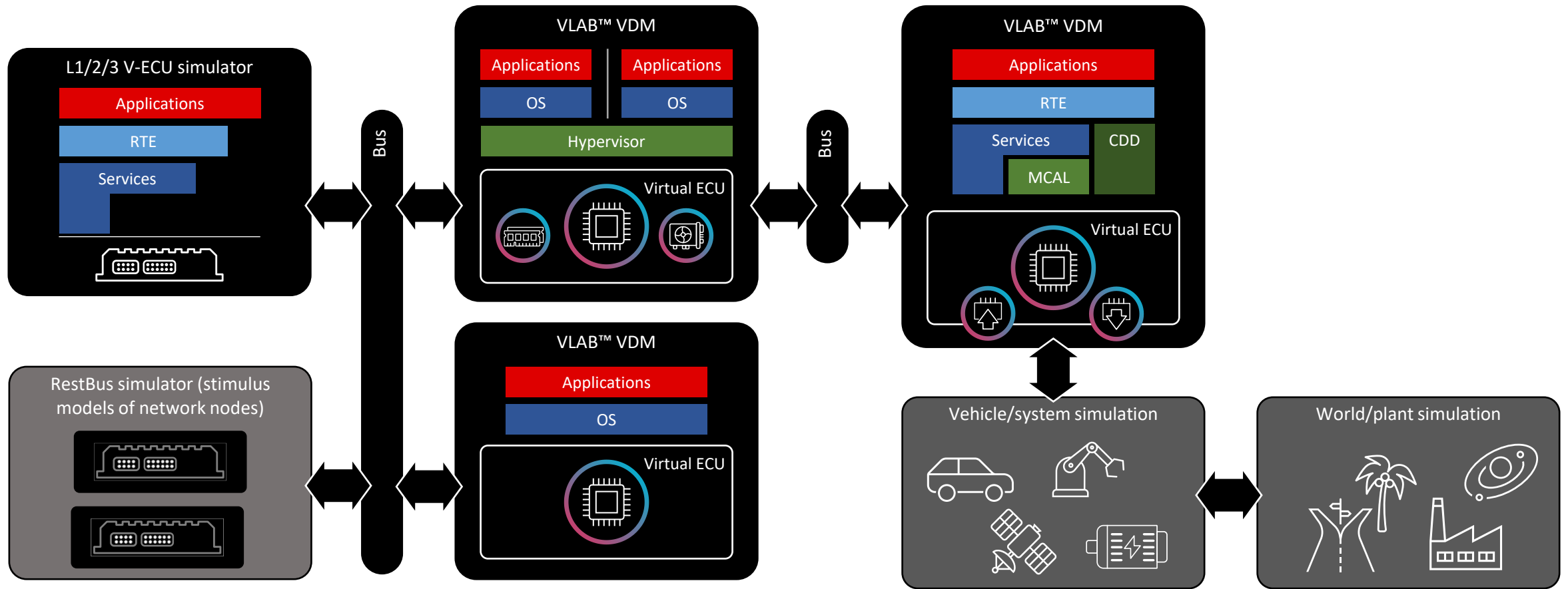
Performance Analysis of Federated Simulations using the Open-Source SIL Kit Library

Jakob Engblom, Cadence (VLAB Works)

engblom@cadence.com



The Goal: Federated Simulation

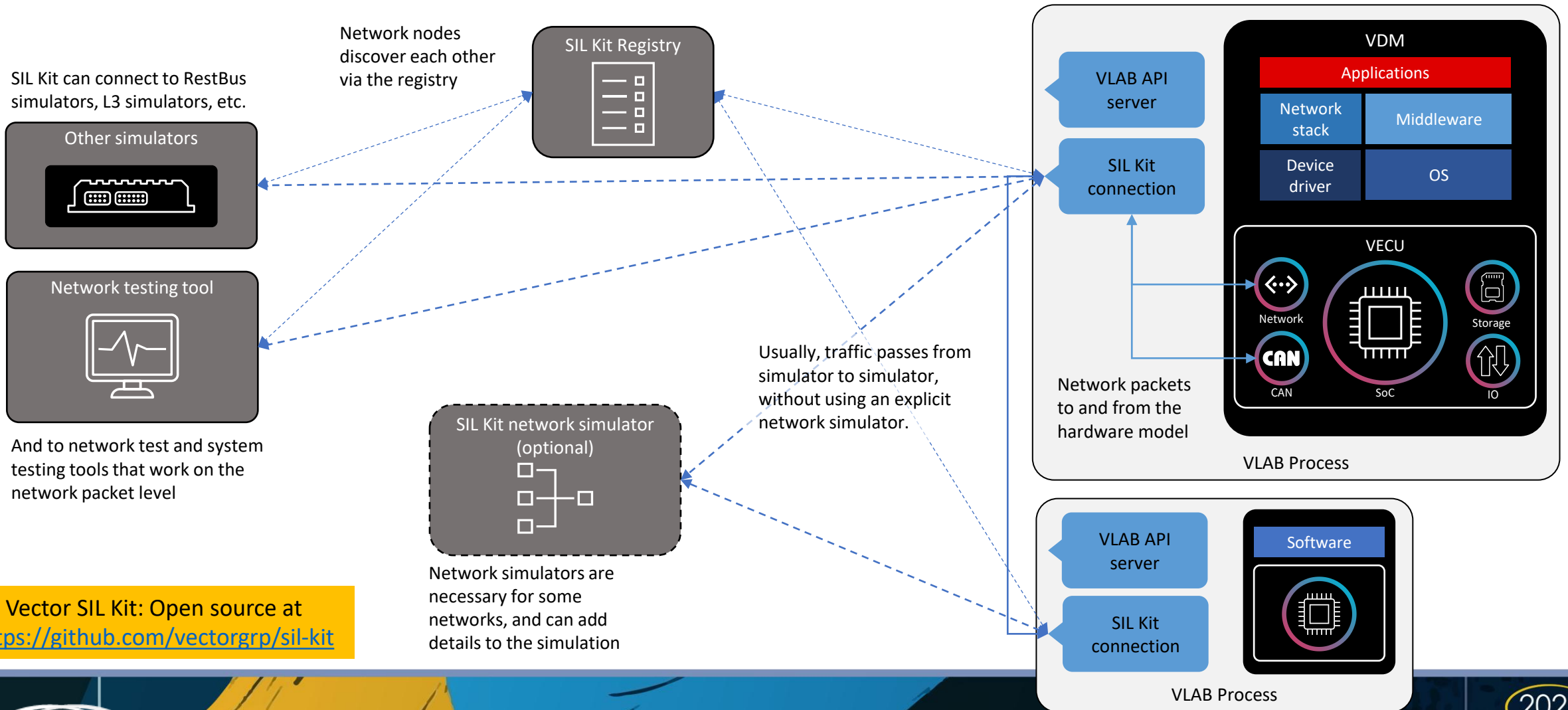




Networking with VLAB

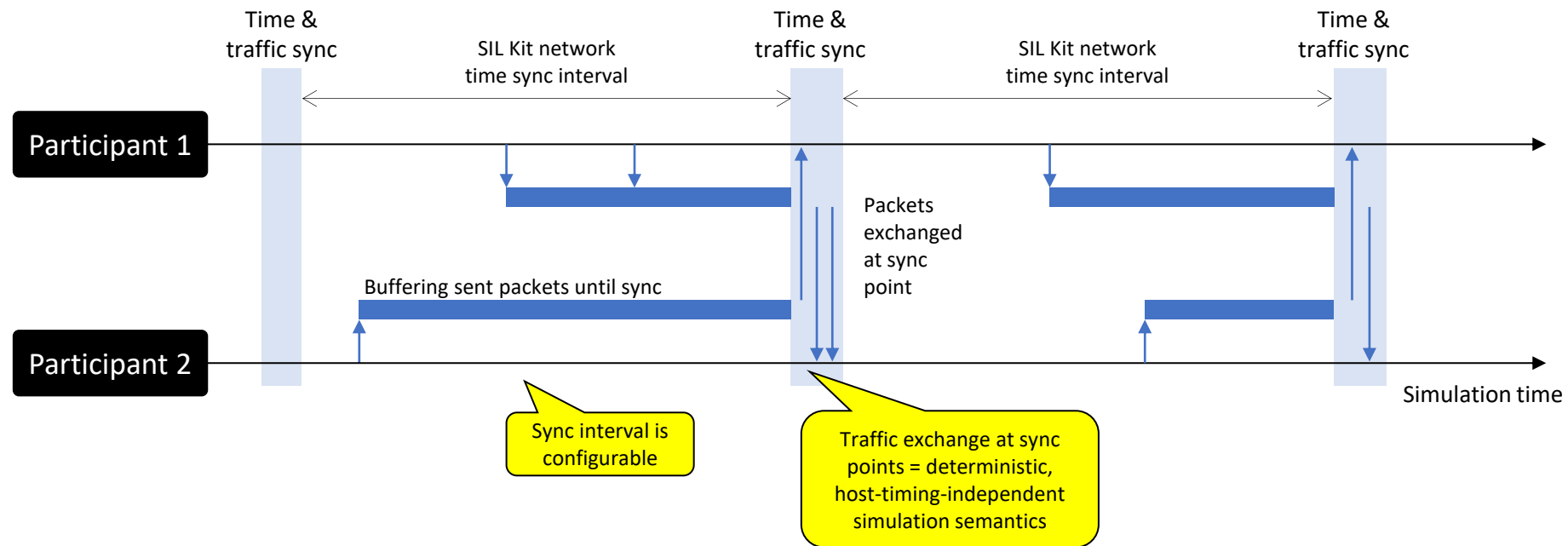


Networking VLAB with SIL Kit



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

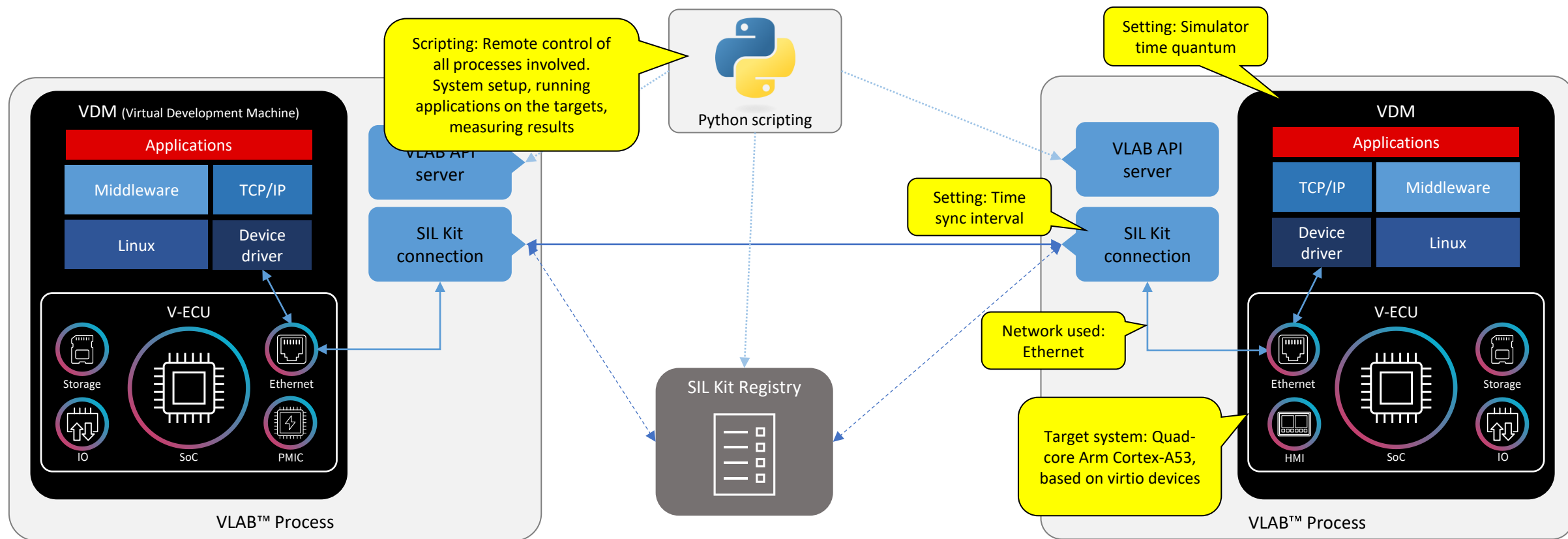
SIL Kit Semantics





Experimental Setup

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()
```

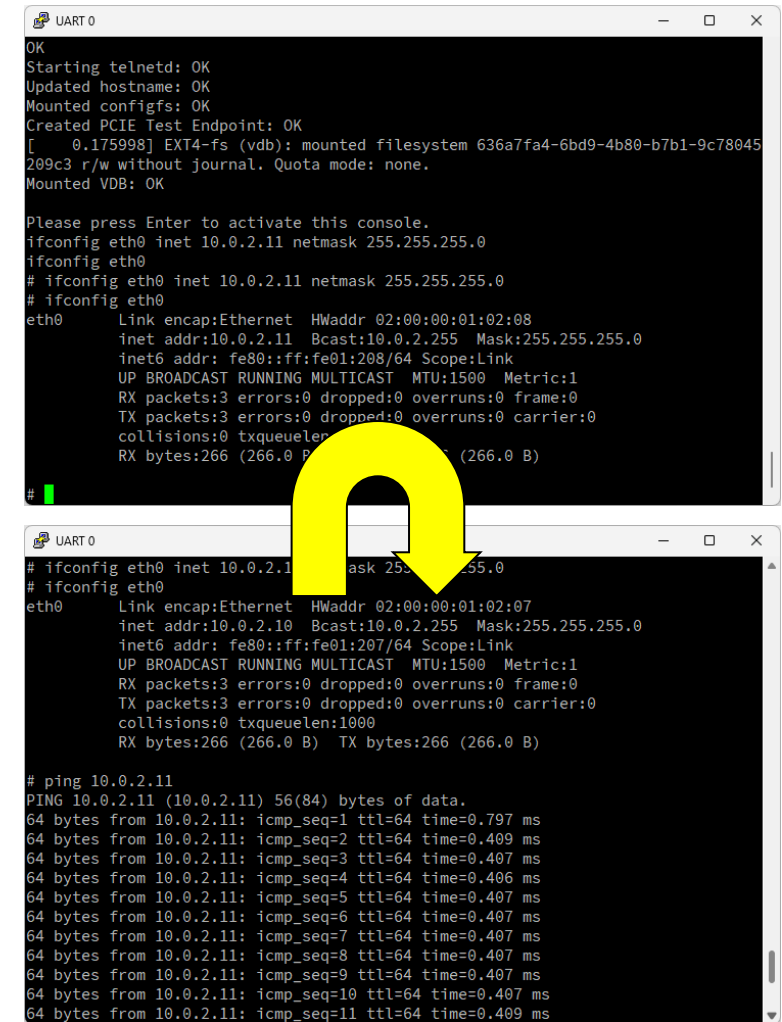



Results



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)



The image shows two terminal windows. The top window displays the output of the 'ifconfig' command for the 'eth0' interface, showing IP address 10.0.2.11 and netmask 255.255.255.0. The bottom window shows the output of the 'ping 10.0.2.11' command, displaying 11 successful ping attempts with round-trip times ranging from 0.407 ms to 0.797 ms. A large yellow arrow points from the top window to the bottom window, indicating a sequence of operations.

```
UART 0
OK
Starting telnetd: OK
Updated hostname: OK
Mounted configs: OK
Created PCIE Test Endpoint: OK
[ 0.175998] EXT4-fs (vdb): mounted filesystem 636a7fa4-6bd9-4b80-b7b1-9c78045
209c3 r/w without journal. Quota mode: none.
Mounted VDB: OK

Please press Enter to activate this console.
ifconfig eth0 inet 10.0.2.11 netmask 255.255.255.0
ifconfig eth0
# ifconfig eth0 inet 10.0.2.11 netmask 255.255.255.0
# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 02:00:00:01:02:08
          inet addr:10.0.2.11  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::ff:fe01:208/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:266 (266.0 B)  TX bytes:266 (266.0 B)

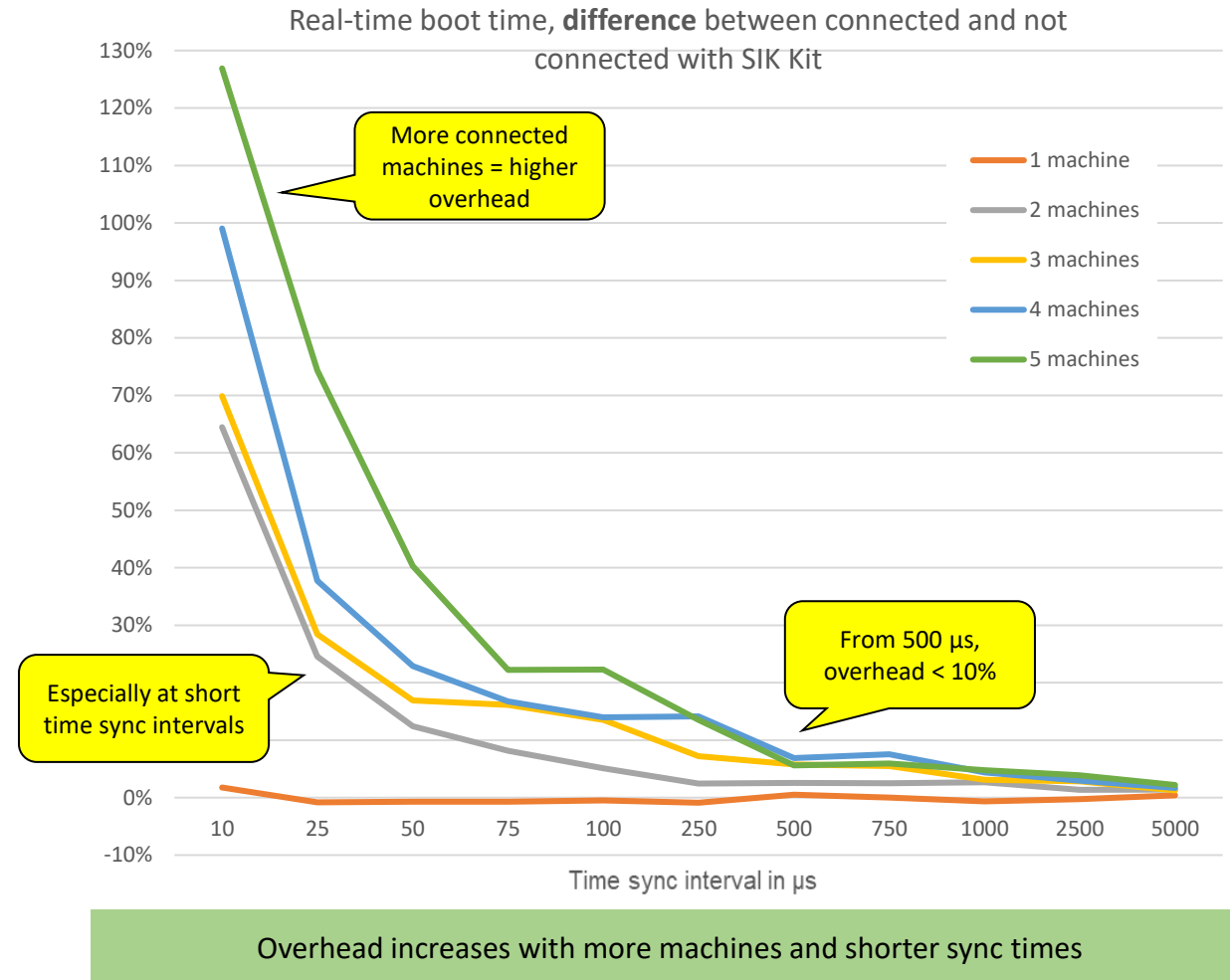
# ifconfig eth0 inet 10.0.2.11 netmask 255.255.255.0
# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 02:00:00:01:02:07
          inet addr:10.0.2.10  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::ff:fe01:207/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:266 (266.0 B)  TX bytes:266 (266.0 B)

# ping 10.0.2.11
PING 10.0.2.11 (10.0.2.11) 56(84) bytes of data.
64 bytes from 10.0.2.11: icmp_seq=1 ttl=64 time=0.797 ms
64 bytes from 10.0.2.11: icmp_seq=2 ttl=64 time=0.409 ms
64 bytes from 10.0.2.11: icmp_seq=3 ttl=64 time=0.407 ms
64 bytes from 10.0.2.11: icmp_seq=4 ttl=64 time=0.406 ms
64 bytes from 10.0.2.11: icmp_seq=5 ttl=64 time=0.407 ms
64 bytes from 10.0.2.11: icmp_seq=6 ttl=64 time=0.407 ms
64 bytes from 10.0.2.11: icmp_seq=7 ttl=64 time=0.407 ms
64 bytes from 10.0.2.11: icmp_seq=8 ttl=64 time=0.407 ms
64 bytes from 10.0.2.11: icmp_seq=9 ttl=64 time=0.407 ms
64 bytes from 10.0.2.11: icmp_seq=10 ttl=64 time=0.407 ms
64 bytes from 10.0.2.11: icmp_seq=11 ttl=64 time=0.409 ms
```

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



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

```
Windows PowerShell
(venv-cortex-a) PS C:\workspace\Experiments\Cortex-A\Cortex-A-SILKit> python .\run_experiment_iperf.py

>>> Run: 1/220 <<< (2025-09-23, Tue, 16:07)
Params: quantum: 1_000_000 | sil kit interval: 10_000_000 | (Terminals)
Running sil-kit-registry
Running sil-kit-system-controller
Elaborating A53 Linux (0)
Elaborating A53 Linux (1)
Listening for "Please press Enter to activate this console." on machine #1
Boot time: 5.462515400024131s 0.2338409435s
Running iperf experiment
Listening for "#" on machine #1 (start at 1_400_600_000_000 ps)
```

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

iPerf client running on the other machine

```
UART 0
209c3 r/w without journal. Quota mode: none.
Mounted VDB: OK

Please press Enter to activate this console.
ifconfig eth0 inet 10.0.2.11 netmask 255.255.255.0
ifconfig eth0
# ifconfig eth0 inet 10.0.2.11 netmask 255.255.255.0
# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 02:00:00:01:02:08
          inet addr:10.0.2.11  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::ff:fe01:208/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:266 (266.0 B)  TX bytes:266 (266.0 B)

# iperf -c 10.0.2.10 -f k -t 5
Client connecting to 10.0.2.10, TCP port 5001
TCP window size: 16.0 KByte (default)
[ 1] local 10.0.2.11 port 38384 connected with 10.0.2.10 port 5001
```

Sending traffic to the server and measuring achieved throughput

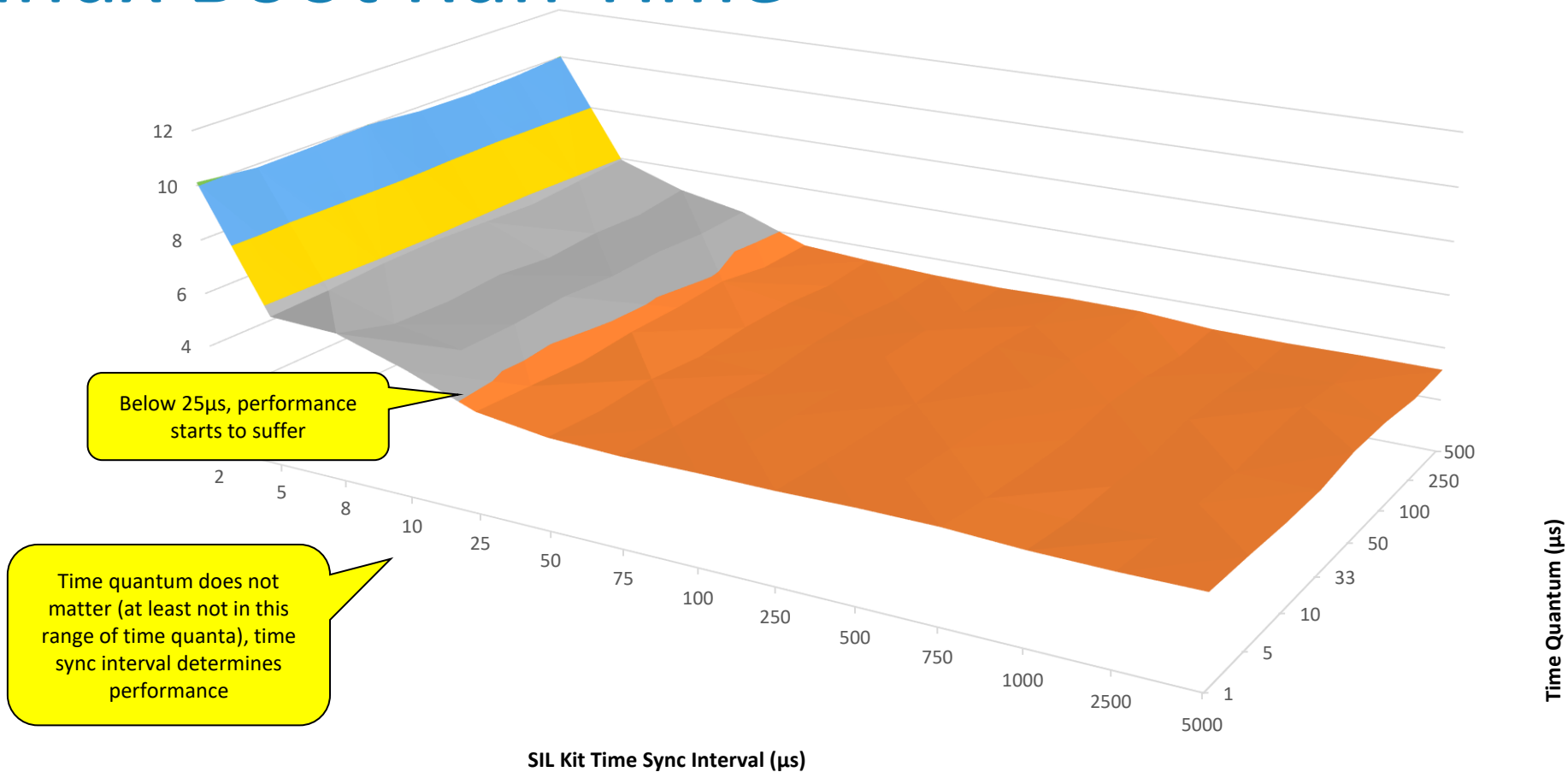
```
UART 0
209c3 r/w without journal. Quota mode: none.
Mounted VDB: OK

Please press Enter to activate this console.
ifconfig eth0 inet 10.0.2.10 netmask 255.255.255.0
ifconfig eth0
# ifconfig eth0 inet 10.0.2.10 netmask 255.255.255.0
# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 02:00:00:01:02:07
          inet addr:10.0.2.10  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::ff:fe01:207/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:266 (266.0 B)  TX bytes:266 (266.0 B)

# iperf -s
Server listening on TCP port 5001
TCP window size: 128 KByte (default)
[ 1] local 10.0.2.10 port 5001 connected with 10.0.2.11 port 38384
```

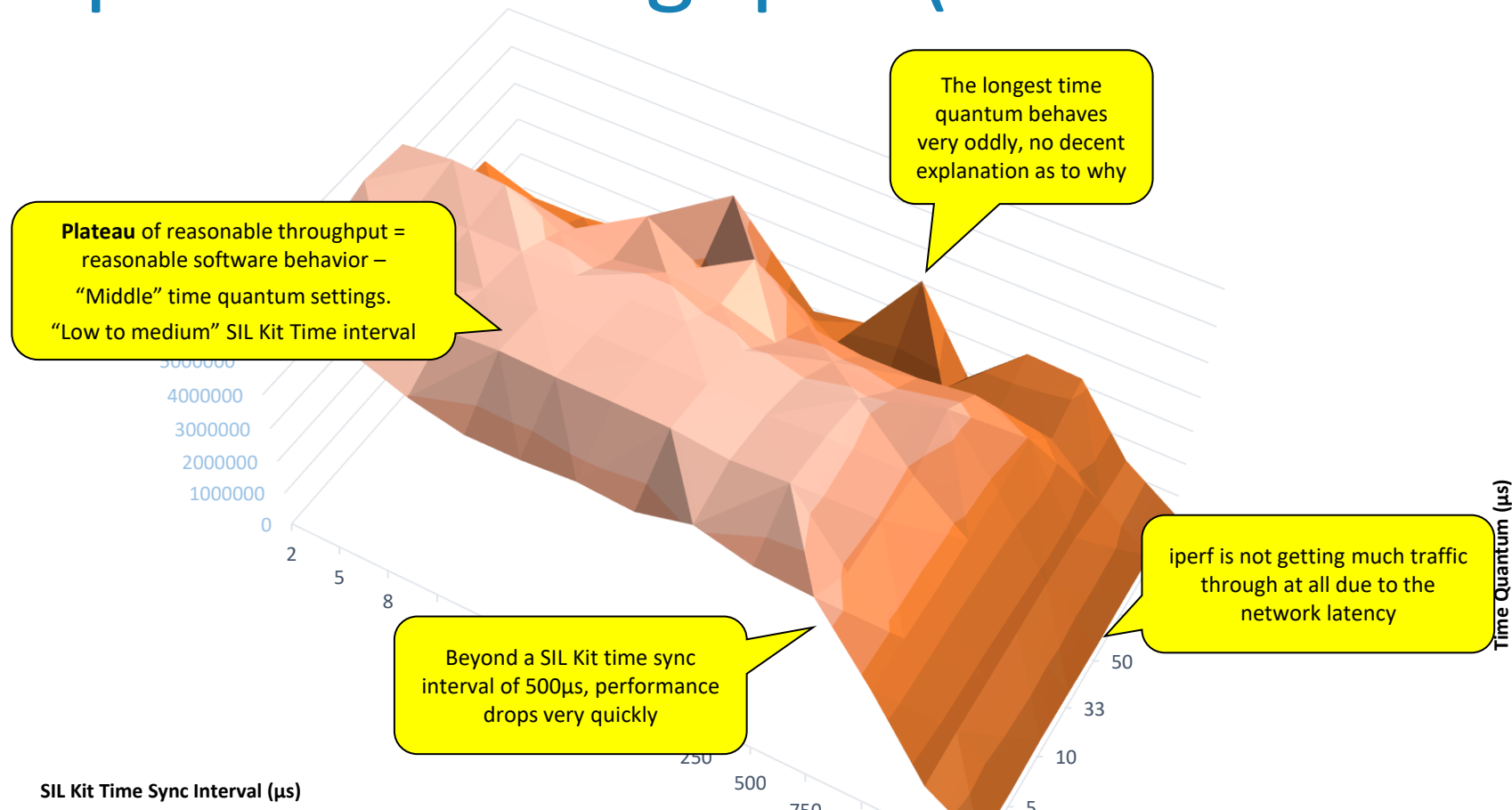
iPerf server running on one machine

Linux Boot Run Time



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

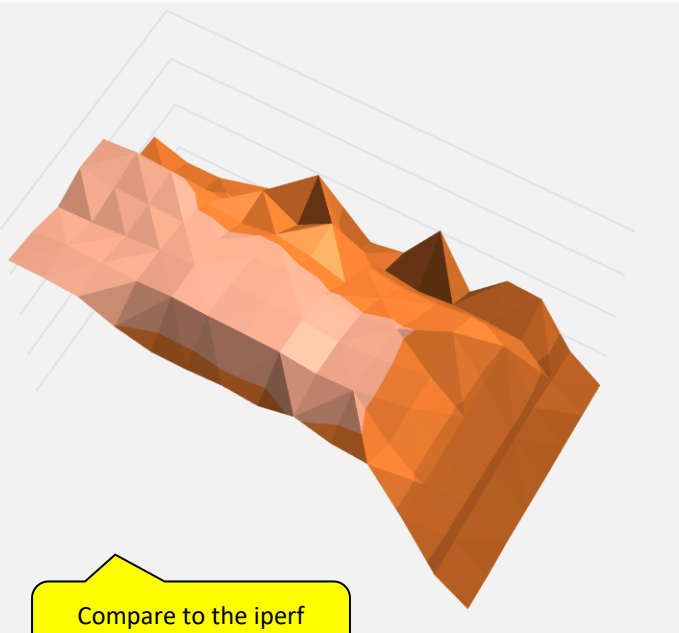
iPerf-Reported Throughput (“Performance”)



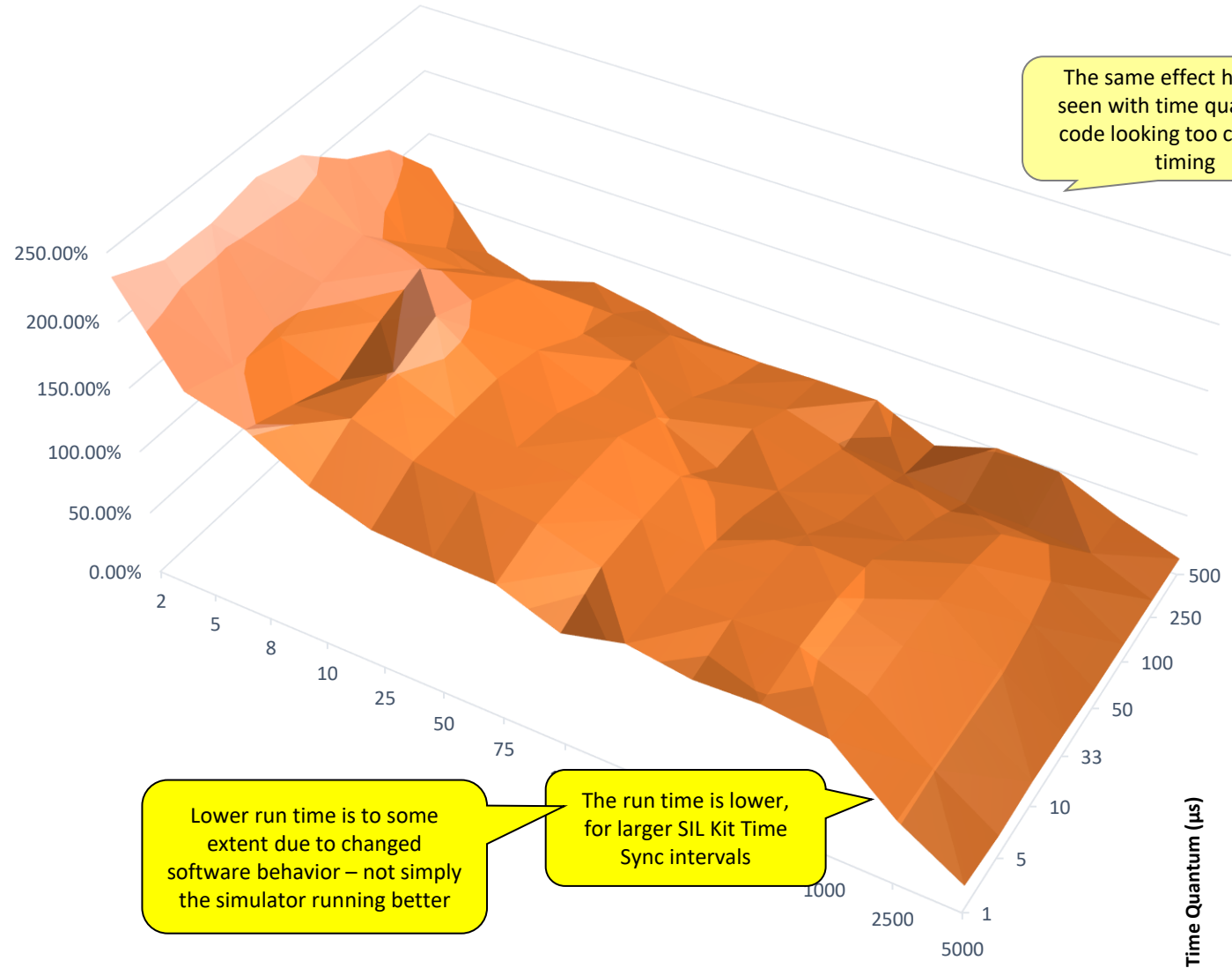
The software-visible behavior varies with the time settings, in sometimes non-obvious ways

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

iPerf Run Time



Compare to the iperf throughput profile



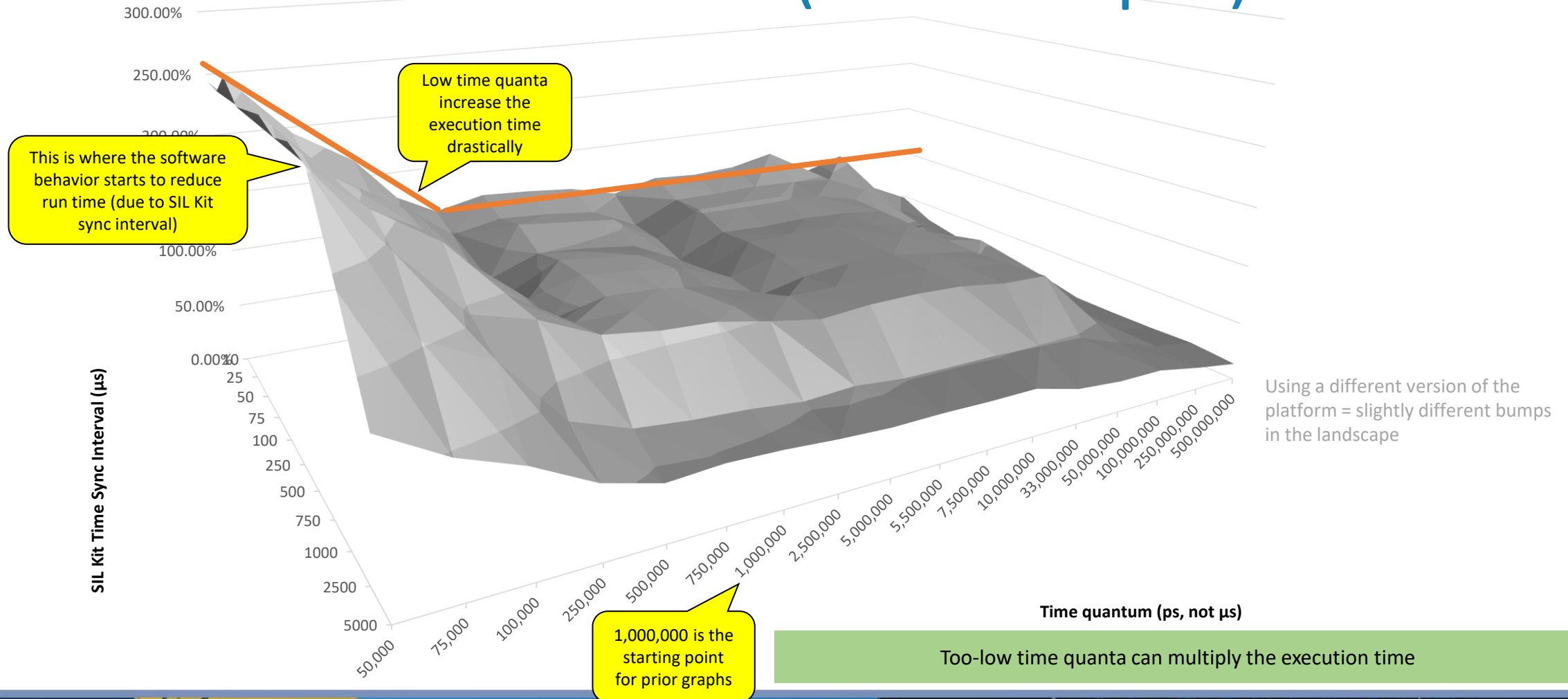
The same effect has been seen with time quanta and code looking too closely at timing

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

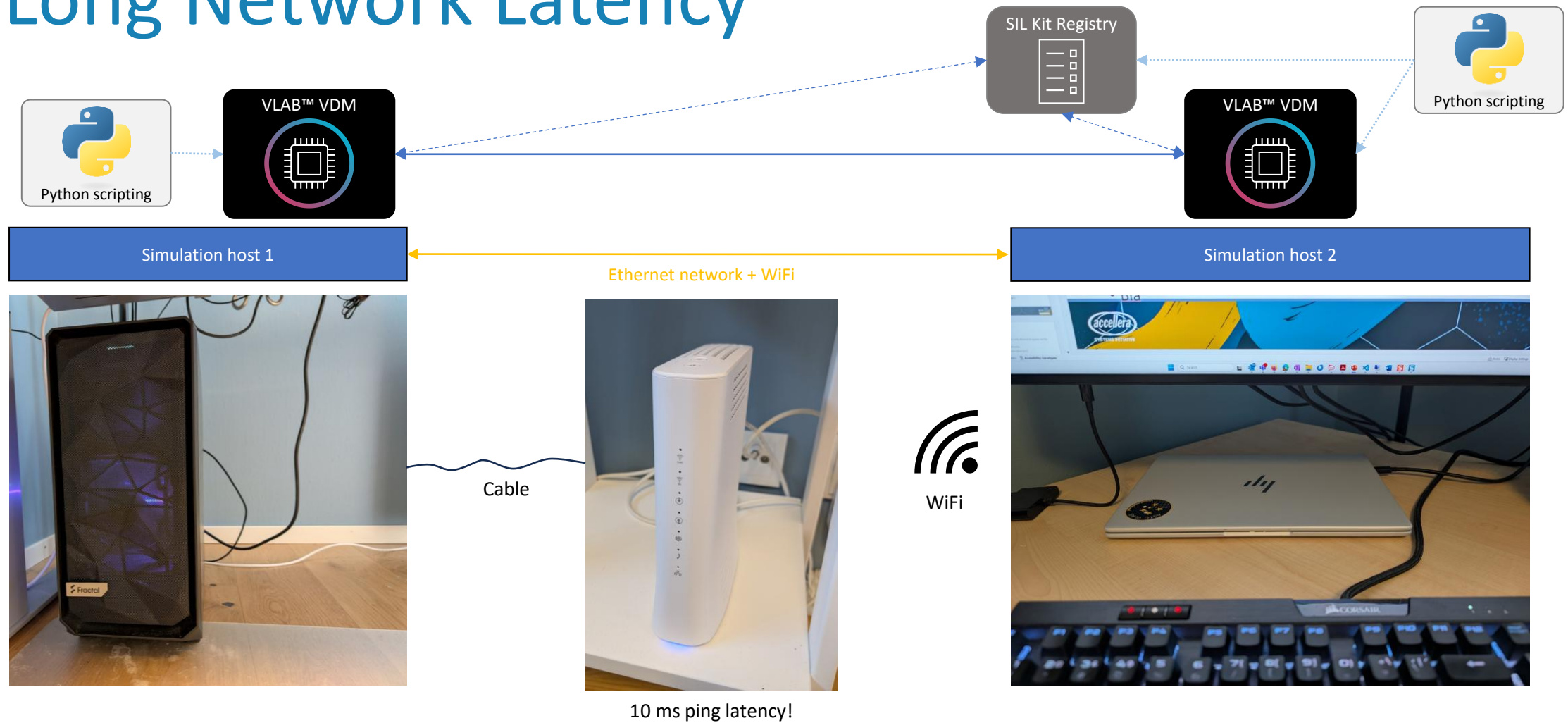
Extended iPerf Run Time (Not in Paper)



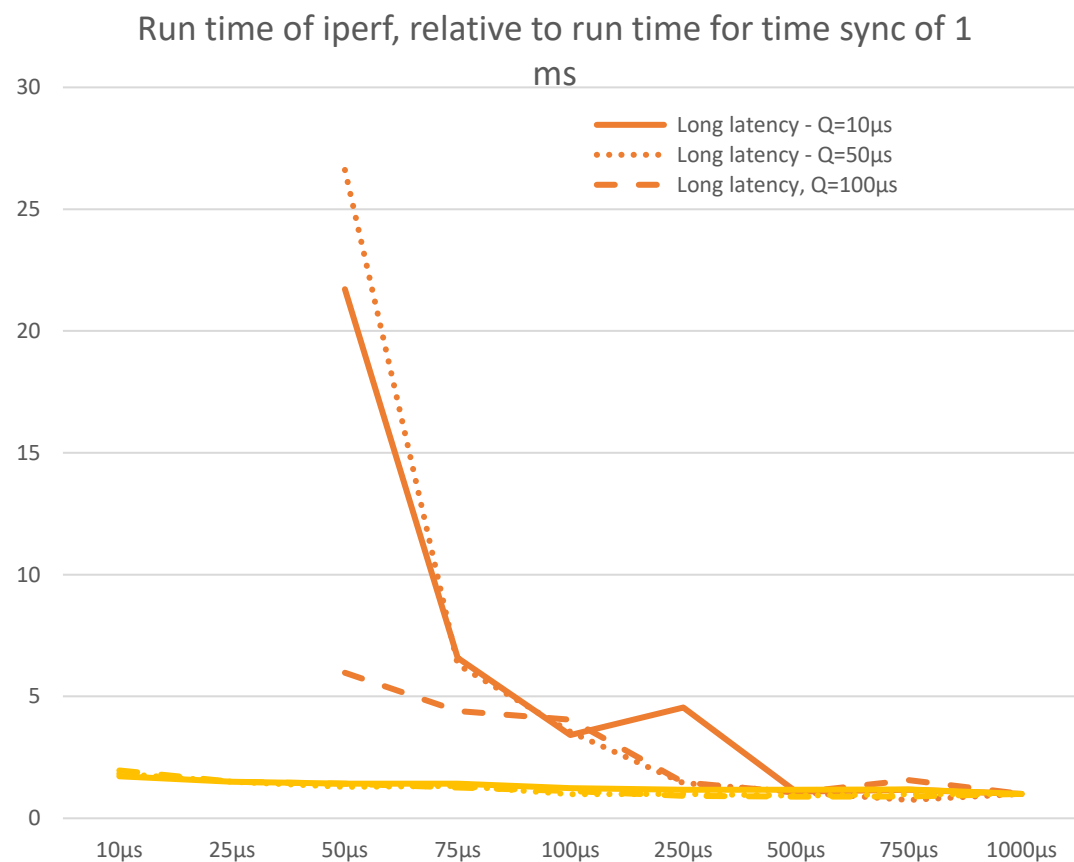


Distributed Simulation

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



Conclusions

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)



Questions?





Backups

iPerf Throughput

