A comparative analysis of Precision Time Protocol in native, virtual machines and container-based environments for consolidating automotive workloads

Speaker: Ong Boon Leong boon.leong.ong@intel.com

Co-authors: Anil Kumar anil.n.kumar@intel.com
Usman Sarwar usman.sarwar@intel.com

2018 IEEE-SA Ethernet & IP @ Automotive Technology Day

9-10 October 2018 London, UK

Internet of Things Group

Agenda

- Overview of Linuxptp
- PTP Synchronization Scenarios:-
  - Inter Native Machine
  - Inter Virtual Machine
  - VM to Local Machine
  - VM to Remote Machine
- ACRN Project: Consolidating Automotive Workloads

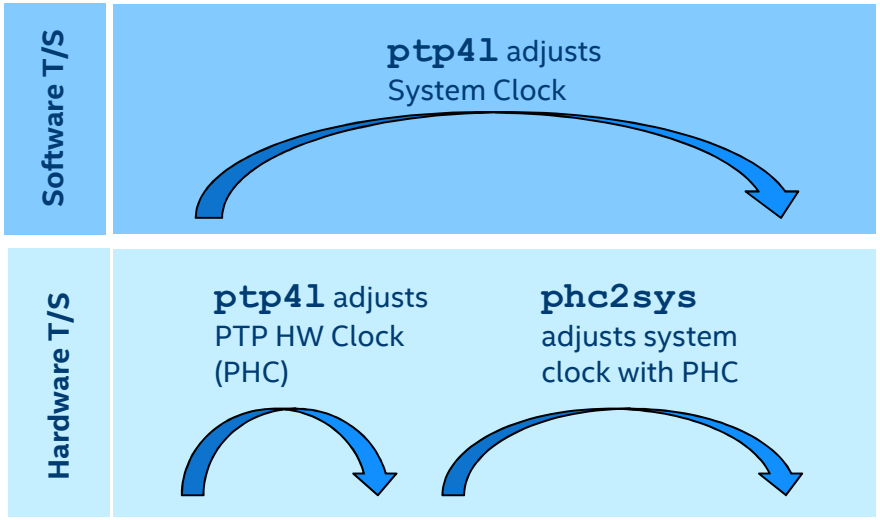
Overview of Linuxptp Project

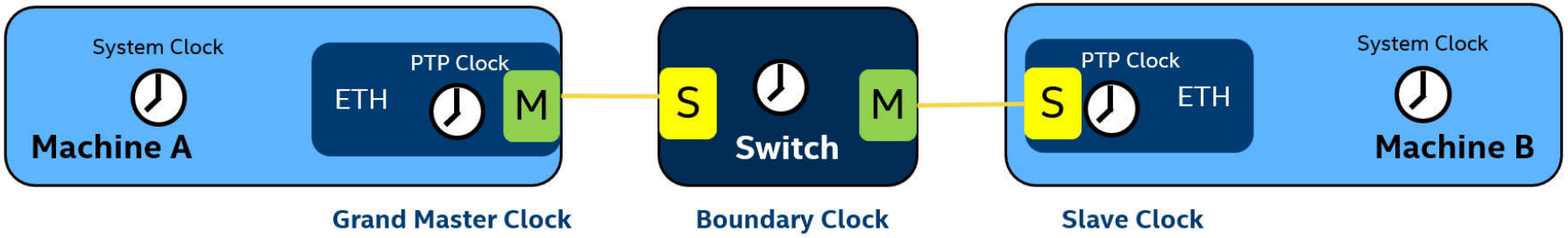
PTP4L & PHC2SYS

Linuxptp Project provides software daemons:

- **`ptp4l`**
  - PTP Message over L2, UDP (IPv4 or IPv6)
  - Hardware & Software Timestamping
  - Synchronize distributed PHC clocks
  - IEEE1588 and compatible with IEEE 802.1AS gPTP daemon (daemon_cl of Avnu)

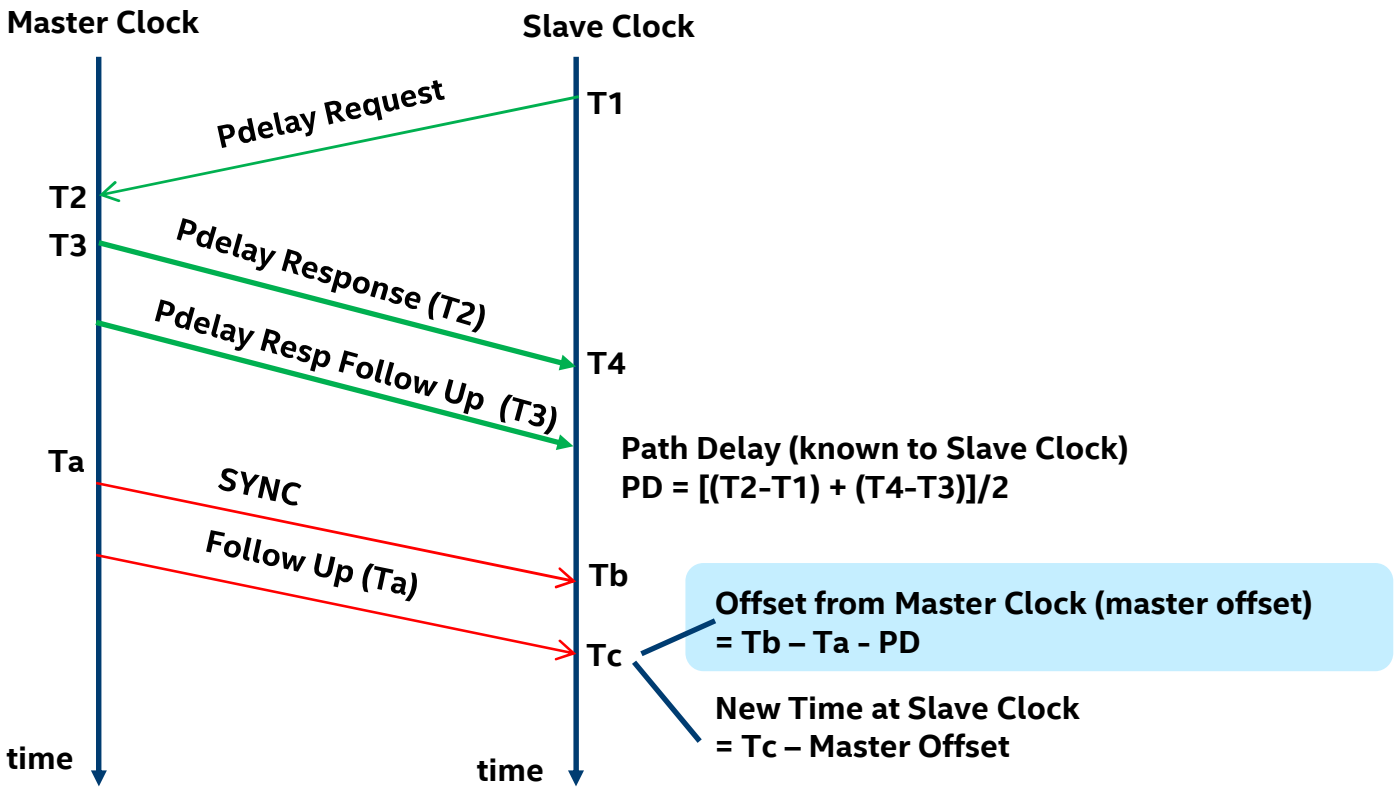
- **`phc2sys`**
  - Synchronize System Clock (CLOCK_REALTIME) to PTP Hardware Clock (PHC)





Overview of Linuxptp Project

OFFSET FROM MASTER CLOCK (MASTER OFFSET)



Overview of Linuxptp Project

PTP4L LOG AT SLAVE CLOCK

```
$ ptp4l -i <eth devname> -H|S -2|4|6 -s -m
```

```
ptp4l[61720.627]: selected /dev/ptp0 as PTP clock
ptp4l[61720.628]: port 1: INITIALIZING to LISTENING on INIT_COMPLETE
ptp4l[61720.628]: port 0: INITIALIZING to LISTENING on INIT_COMPLETE
ptp4l[61722.021]: port 1: new foreign master e8ea6a.fffe.0938e8-1
ptp4l[61726.022]: selected best master clock e8ea6a.fffe.0938e8
ptp4l[61726.022]: port 1: LISTENING to UNCALIBRATED on RS_SLAVE
ptp4l[61728.355]: master offset      -1207 s0 freq   +2406 path delay        -7
ptp4l[61729.355]: master offset      -1287 s2 freq   +2326 path delay        -6
ptp4l[61729.356]: port 1: UNCALIBRATED to SLAVE on MASTER_CLOCK_SELECTED
ptp4l[61730.356]: master offset      -1284 s2 freq   +1042 path delay        -7
ptp4l[61731.356]: master offset          5 s2 freq   +1946 path delay       -11
ptp4l[61732.356]: master offset        408 s2 freq   +2350 path delay       -32
ptp4l[61733.356]: master offset        386 s2 freq   +2451 path delay       -32
ptp4l[61734.356]: master offset        241 s2 freq   +2422 path delay       -11
ptp4l[61735.356]: master offset        143 s2 freq   +2396 path delay        -7
ptp4l[61736.357]: master offset         83 s2 freq   +2379 path delay        -7
ptp4l[61737.357]: master offset         33 s2 freq   +2354 path delay        -8
ptp4l[61738.357]: master offset          6 s2 freq   +2337 path delay        -8
ptp4l[61739.357]: master offset          0 s2 freq   +2332 path delay       -11
```

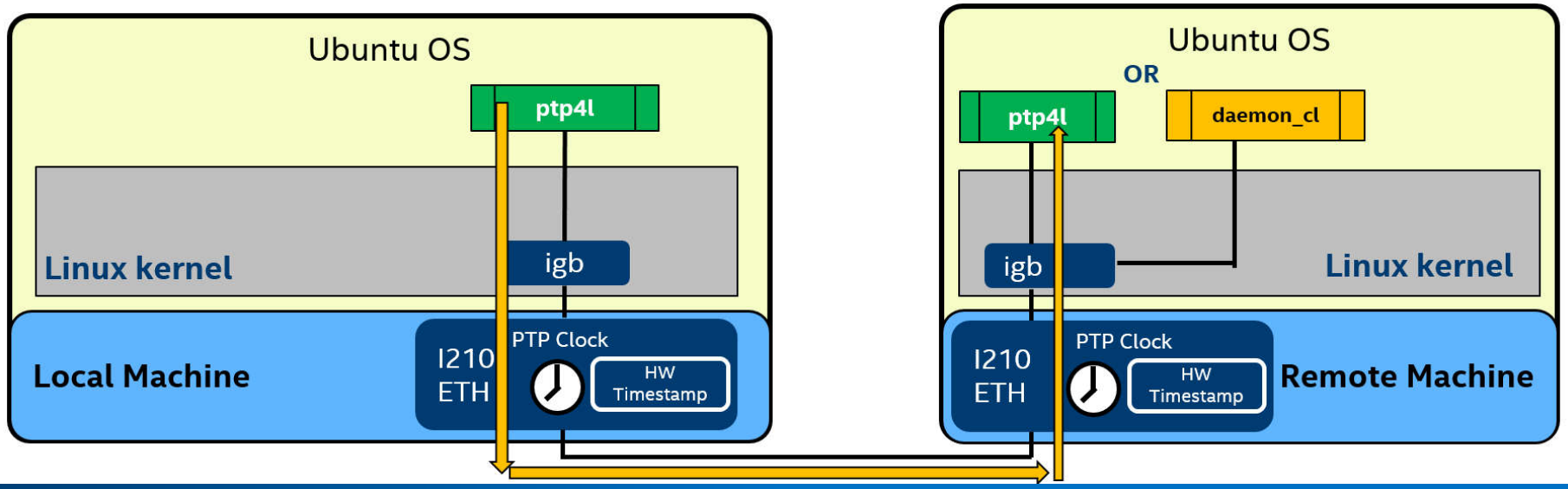
1. **Offset from Master Clock**
2. **S0: unlock**
   S1: clock step
   S2: locked
3. **PHC Frequency Adjustment (pbb)**
4. **Path Delay values between Master and Slave Clocks**

Inter Native Machines

TEST SETUP

Objective:

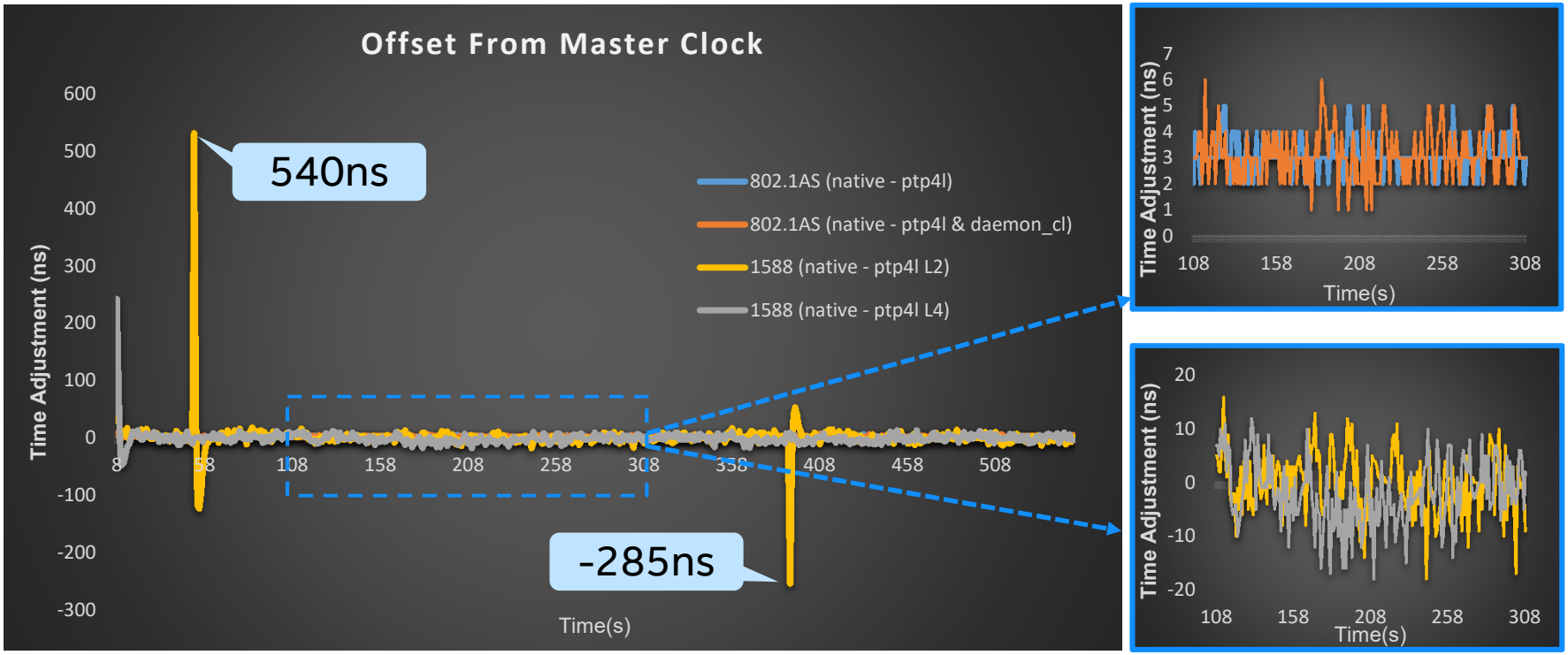
- To obtain baseline PTP synchronization performance for native environment under direct end to end connection.
- PTP message timestamping mode: Hardware Vs Software.
- Cross check ptp4l (linuxptp under gPTP mode) works with daemon_cl (openavnu).



Inter Native Machines

TEST RESULT: HARDWARE TIMESTAMPING (L2 & L4)

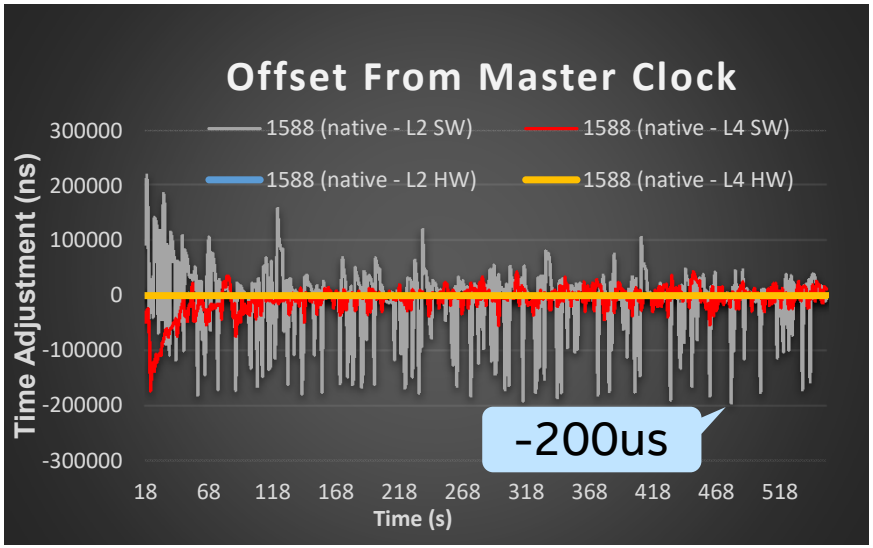
Synch spike: ns	Sync variation: ns

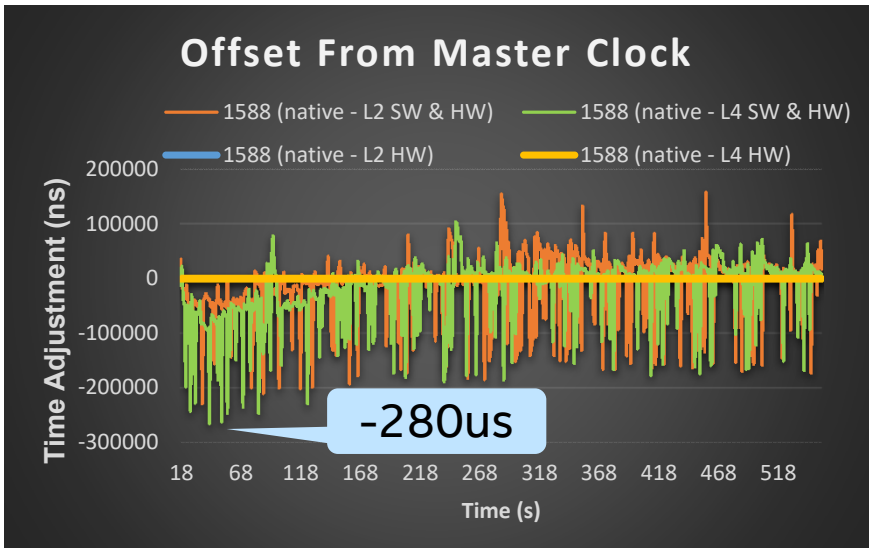


Inter Native Machines

TEST RESULT: SOFTWARE TIMESTAMPING (L2 & L4)

Synch spike: us | Sync variation: us

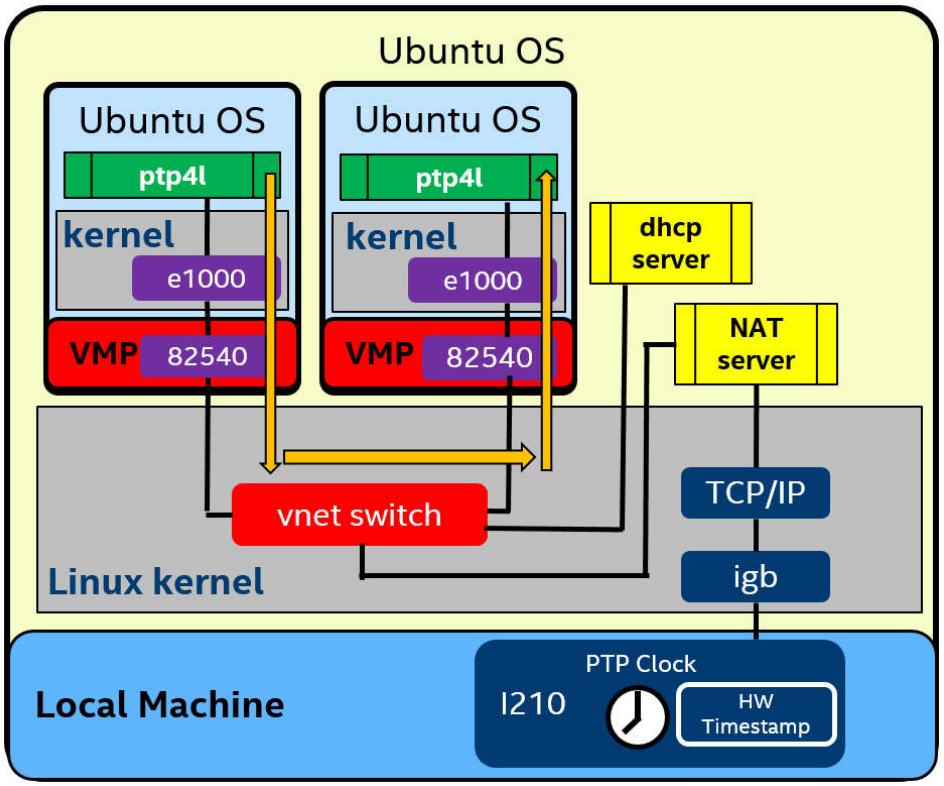




	1588 L2 HW	1588 L4 HW	1588 L2 SW	1588 L4 SW	1588 L2 SW+HW	1588 L4 SW+HW
Mean	0.88	-0.91	-6,068.86	-7,206.24	-12,280.36	-21,737.50
Std. Dev.	28	6	65,881	25,086	63,306	61,330
Max	531	13	217,991	43,634	158,221	104,007
Min	-254	-18	-194,514	-172,642	-228,450	-264,095

Inter Virtual Machines

TEST SETUP: VNET SWITCH + NAT NETWORKING BETWEEN VM'S



Objective:

- To obtain PTP synchronization performance for **Virtual Machine to Virtual Machine** under the same machine.

Facts:

- For VM, we used VMPlayer with "NAT"
- For "VM to VM" is within vnet switch. "VM to Internet" is over NAT server.
- Inside a VM, we see e1000 driver loaded for intel 82540 virtual card.
  - **No PTP hardware time-stamping mode**.
- This configuration supports **L2 & L4 PTP messages**.

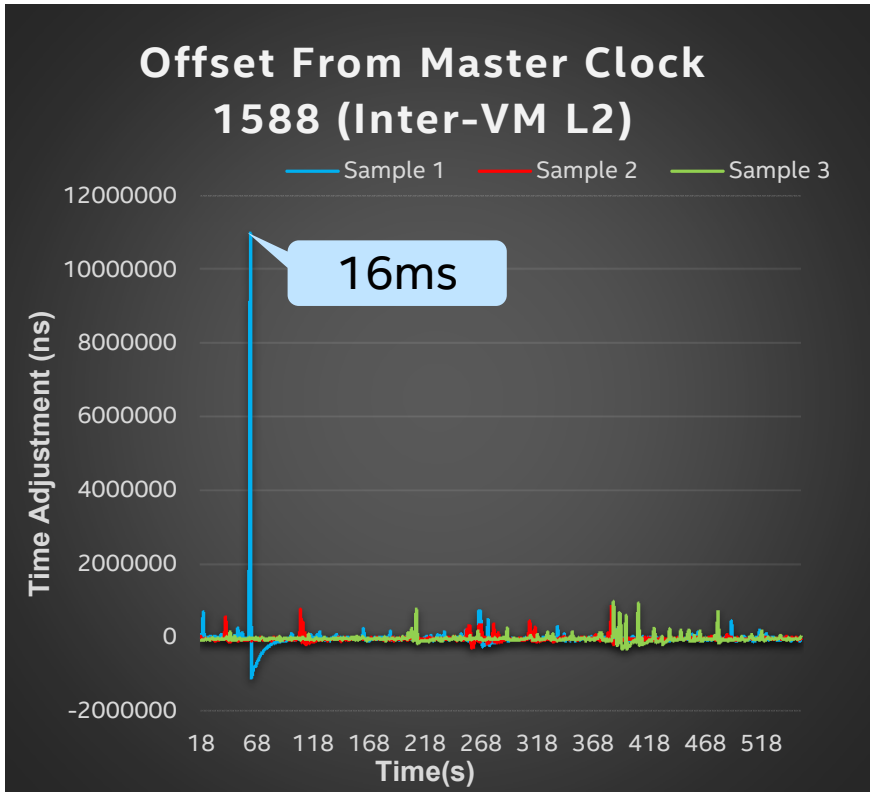
Objective (revised):

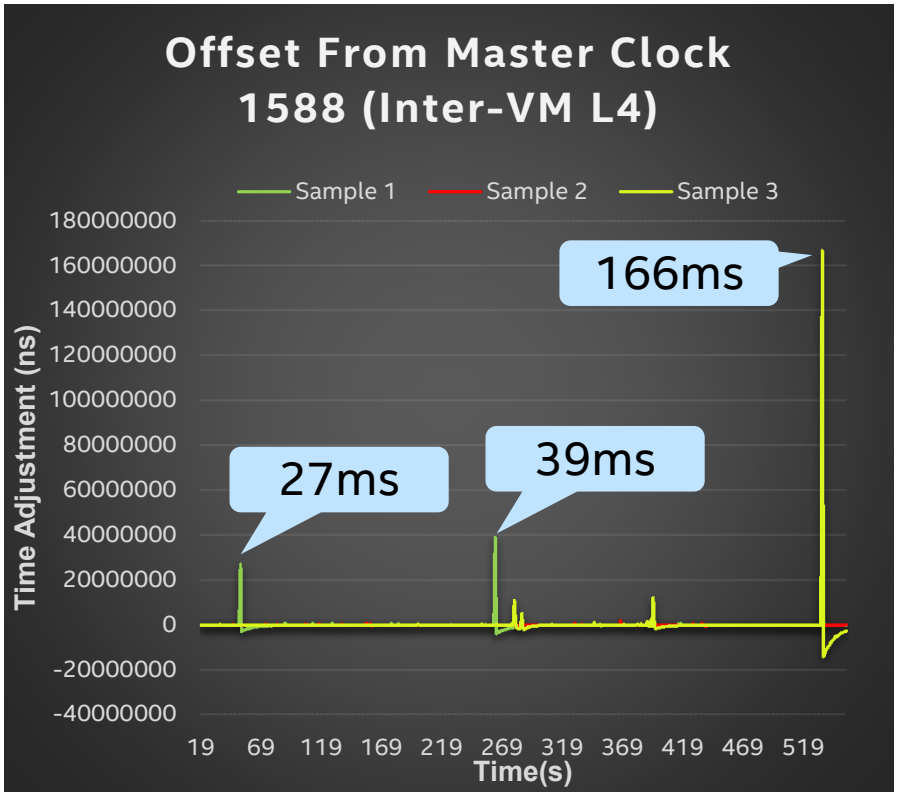
- To obtain default PTP synchronization performance for **VM to VM** under the same machine **for software time-stamping only**.

Inter Virtual Machines

TEST RESULT: SOFTWARE TIMESTAMPING (L2 & L4)

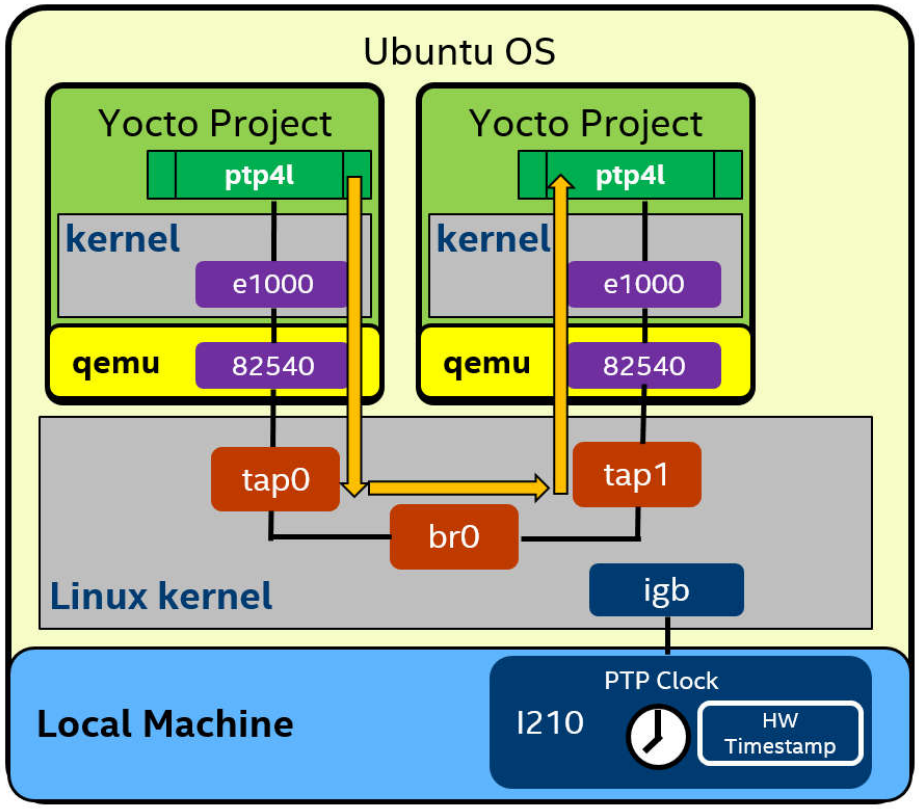
Synch spike: ms | Sync variation: µs





Inter Virtual Machines

BRIDGED NETWORKING (TAP) BETWEEN QEMUS



Objective:

- To obtain PTP synchronization performance for **Qemu to Qemu** under the same machine for **software time-stamping only** over in-kernel L2 software bridge

Facts:

- For Qemu, we used **qemu-kvm** option.
- Inside a Qemu, we see e1000 driver loaded for intel 82540 virtual card (no PTP hardware time-stamping)
- In Host, Qemu is connected to tap/tun driver.
- We connect tap0 and tap1 to br0 (software bridge).
- Under software bridge, L2 PTP message sent out from tap interface contains Ethernet FCS error. So, L2 PTP synchronization fails.

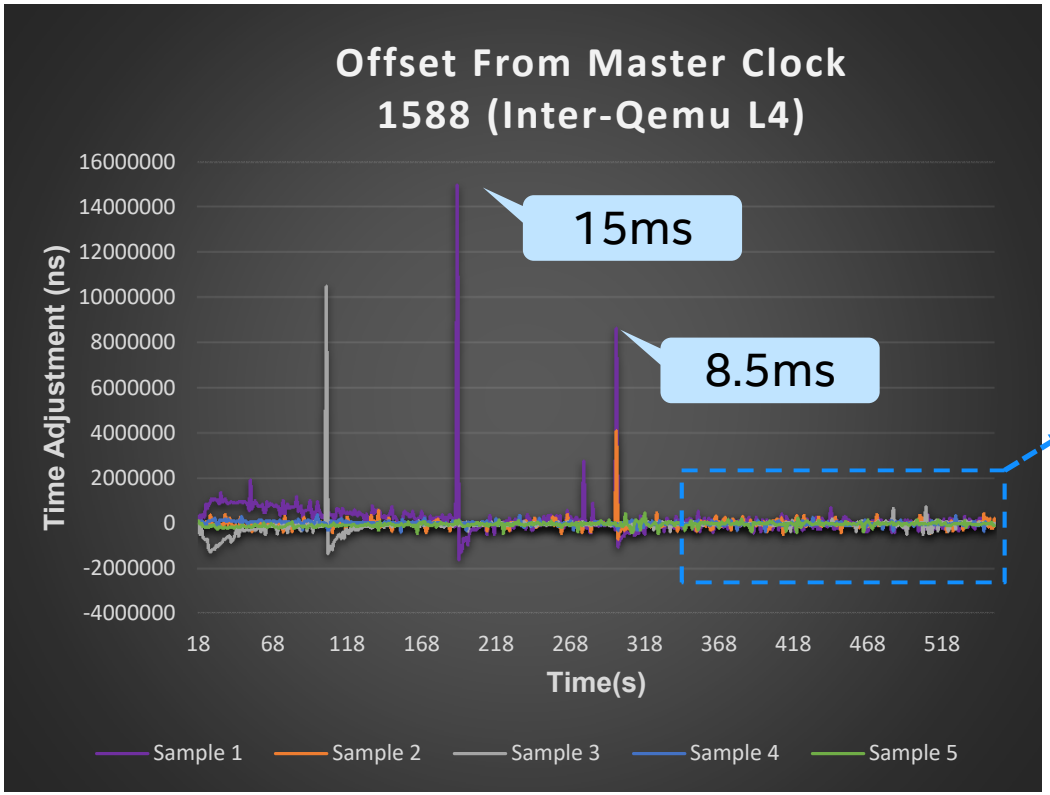
Objective (revised):

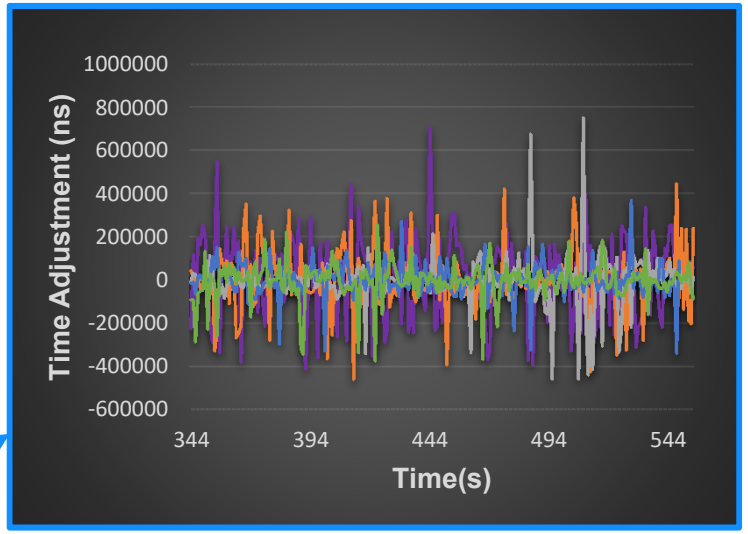
- To obtain default PTP synchronization performance for **Qemu to Qemu** under the same machine for **software time-stamping only** over in-kernel L2 software bridge for **L4 PTP message only**.

Inter Virtual Machines

TEST RESULT: SOFTWARE TIMESTAMPING (L4 ONLY)

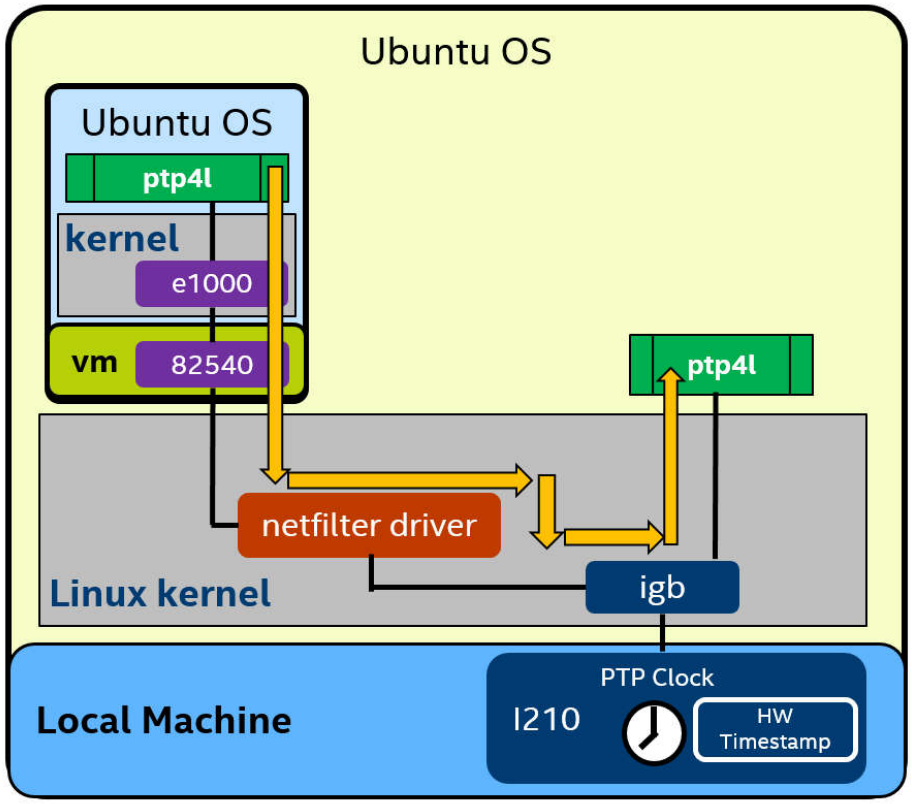
Synch spike: ms | Sync variation: µs





VM to Local Machine

TEST SETUP



Objective:

- To obtain PTP synchronization performance for **Virtual Machine to Host Machine** for **software time-stamping only** over **bridged networking**.
- Configurations:-
  - Grandmaster clock in Host Machine + Slave Clock in VM.
  - Grandmaster clock in VM + Slave Clock in Host Machine.

Facts:

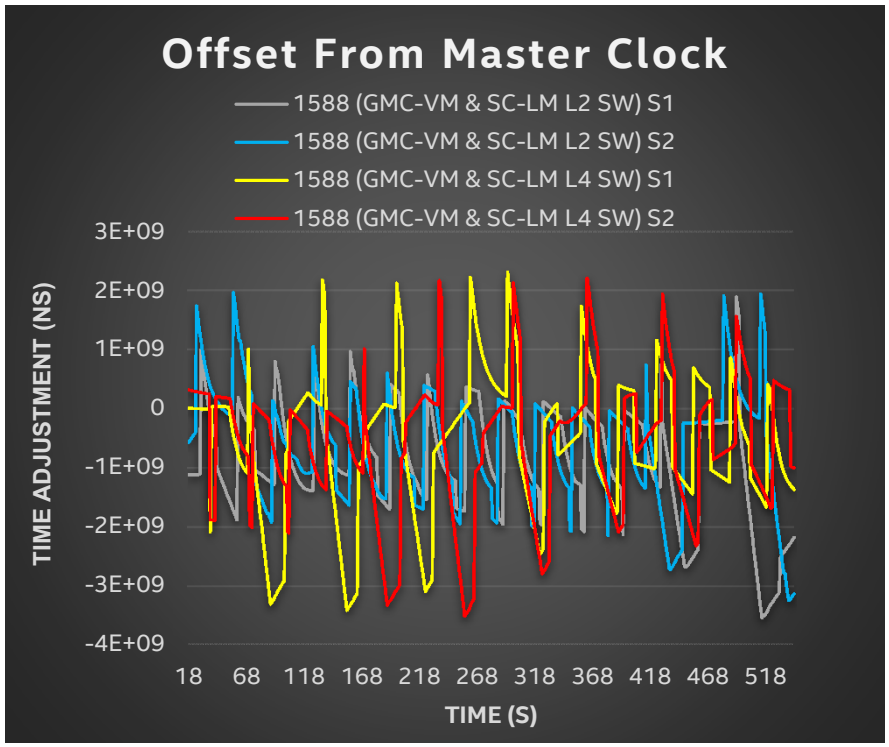
- netfilter intercepts PTP message exchange (multicast) and forward them between VirtualBox and Host Machine.
- This configuration **supports both L2 & L4 PTP messages.**

VM to Local Machine

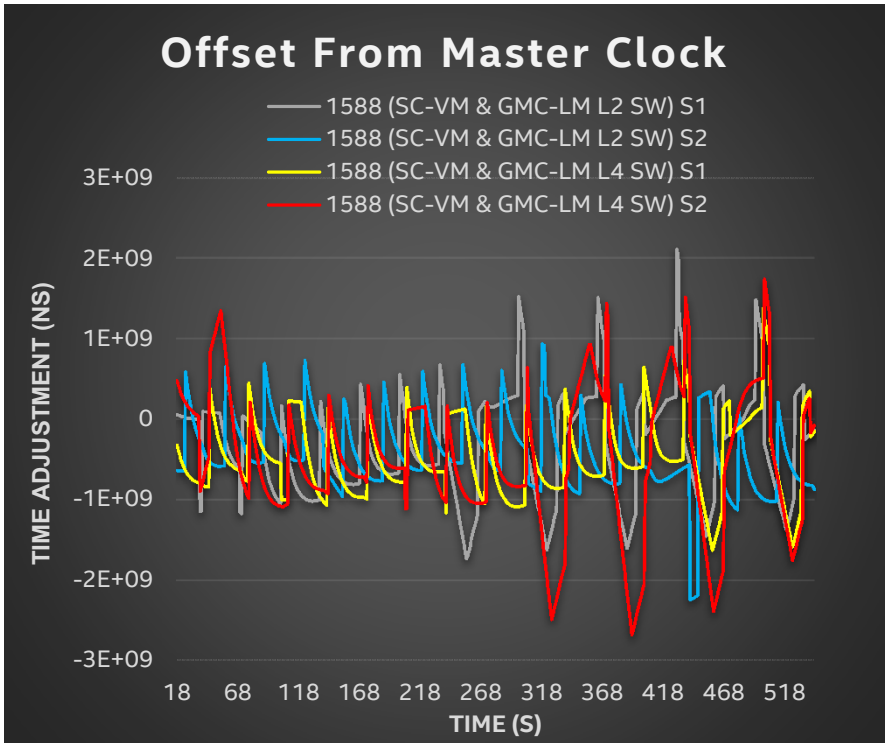
TEST RESULT: SOFTWARE TIMESTAMPING (L2 & L4)

Synch spike: s

Sync variation: s



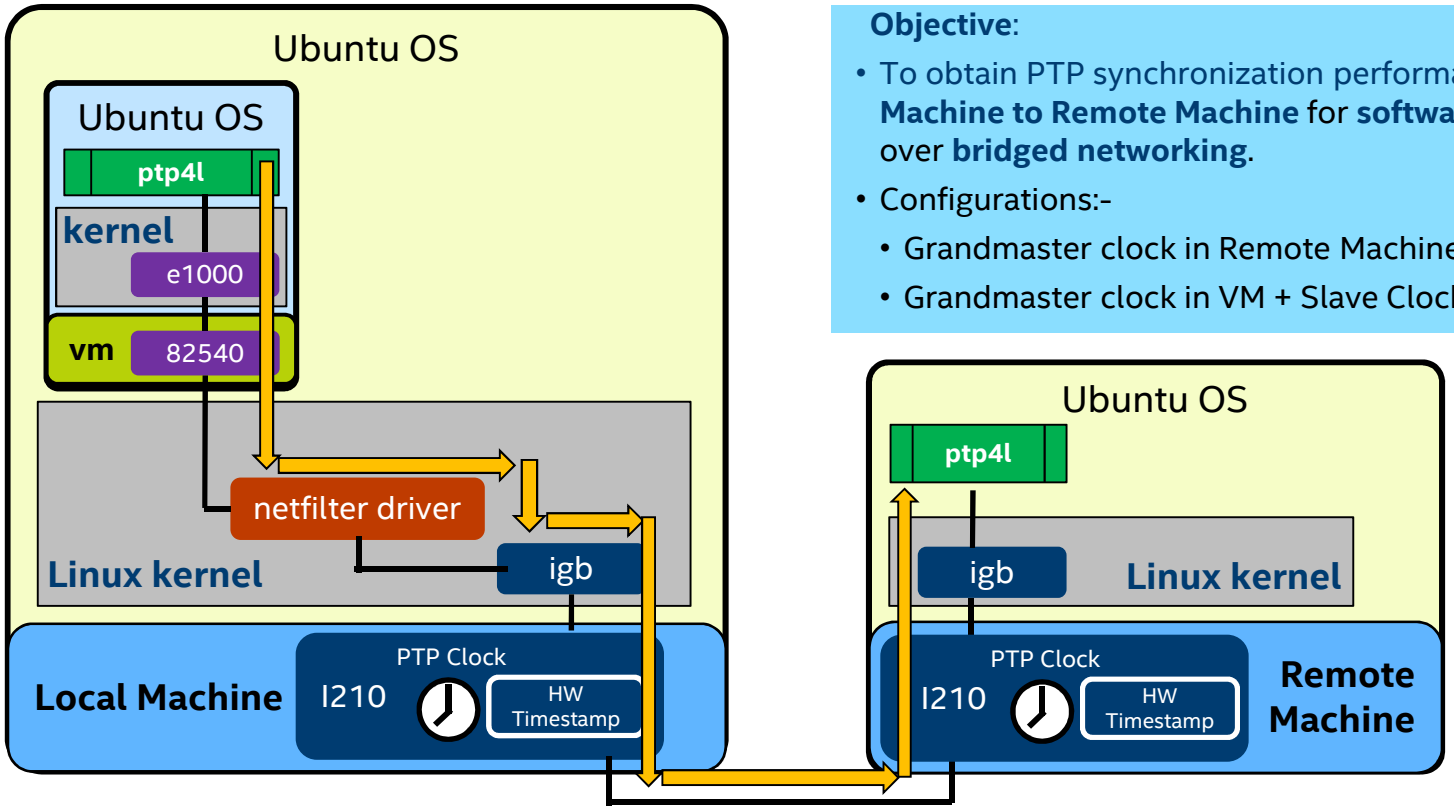
GM at Virtual Machine



GM at Local Host

VM to Remote Machine

TEST SETUP



Objective:

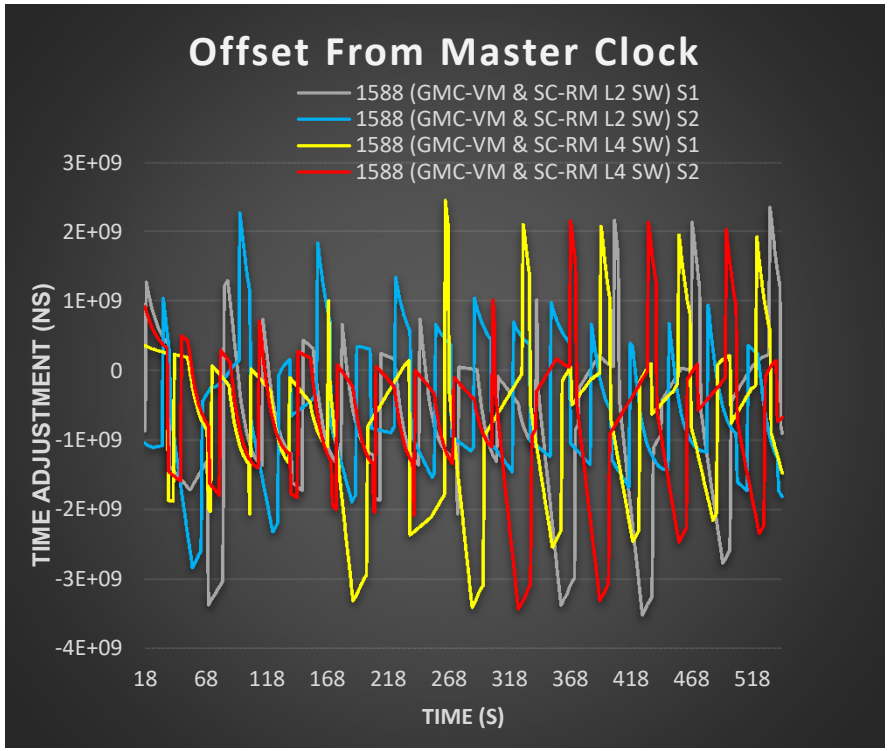
- To obtain PTP synchronization performance for **Virtual Machine to Remote Machine** for **software time-stamping only** over **bridged networking**.
- Configurations:-
  - Grandmaster clock in Remote Machine + Slave Clock in VM.
  - Grandmaster clock in VM + Slave Clock in Remote Machine.

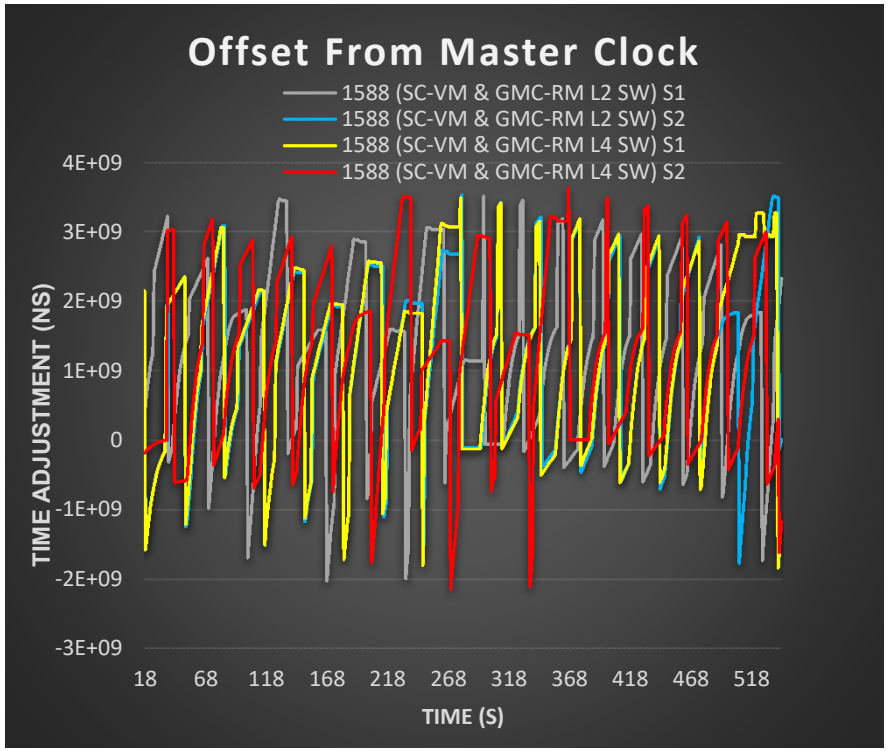
VM to Remote Machine

Synch spike: s

Sync variation: s

TEST RESULT: SOFTWARE TIMESTAMPING (L2 & L4)





Summary of Results

	Sync Variation	Sync Spike/Stray
Inter-machine (HW timestamping)	+/- 20 ns	-200ns to +550ns spike
Inter-machine (SW timestamping)	+/- 200 us	Not Observed
Inter VM (Qemu + L2 bridge)	+/- 400 us	-2ms to +15ms spike
Inter VM (VMP + vnet switch)	+/- 400 us	-1ms to +160ms spike
VM to Local or Remote Machine	Synchronization Fault Error	

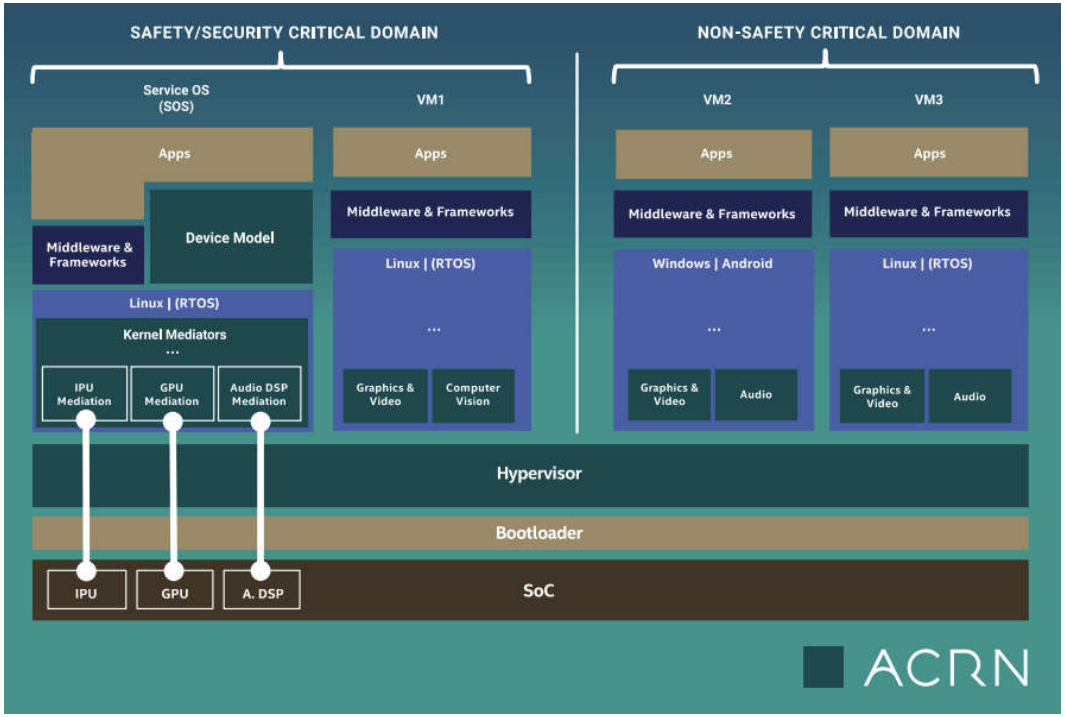
Key Points:

- Hardware time-stamping gives best time sync accuracy.
- Virtual switch introduces additional PTP time synchronization jitter.

ACRN Project: Consolidating Automotive Workloads

Internet of Things Group

Overview of ACRN Project



ACRN Project Value Proposition

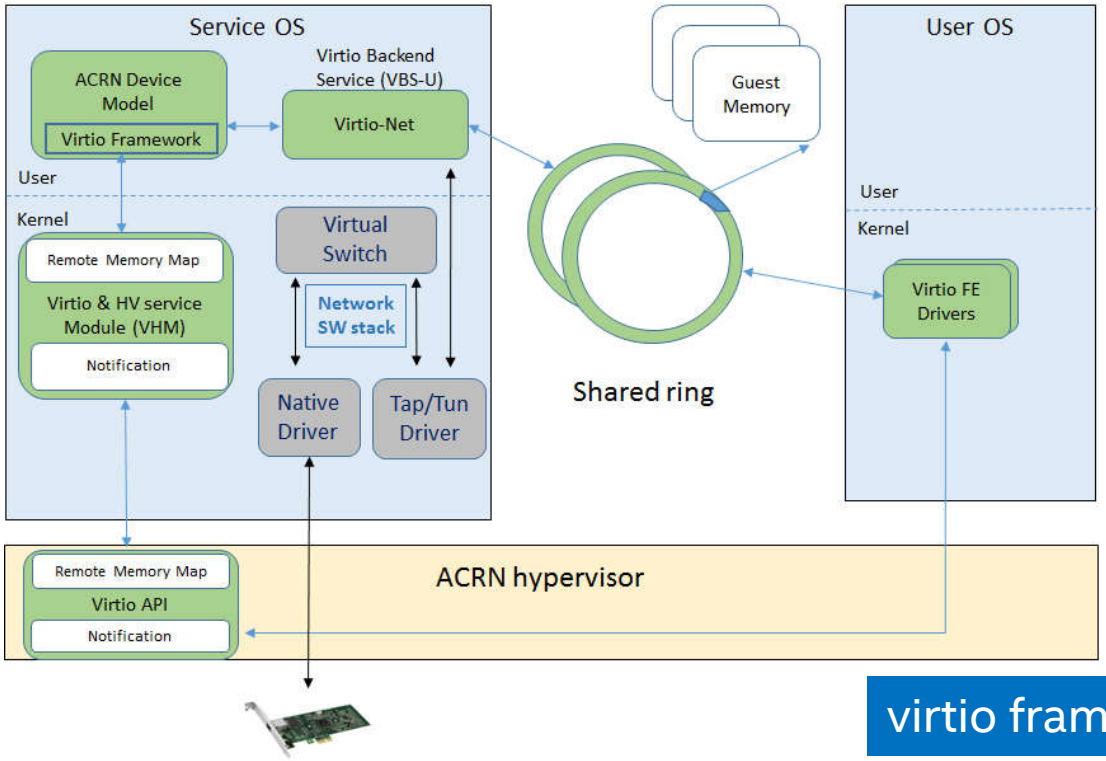
- **Small Code Size / Lightweight**
- **Open Source Project**
- **Real Time**
- **Safety Criticality In Mind**
- **Support Multi OS**

ACRN Hypervisor

- **Type 1 Hypervisor (bare-metal)**
- **Service OS (VM0) [Dom0 in Xen]**
- **User OS (VM1) [DomU in Xen]**
- **Reference I/O Mediation**

https://projectacrn.org/wp-content/uploads/sites/59/2018/07/ACRN-Overview_v14_Web.pdf

Ethernet Networking between VMs: virtio-based I/O



Ethernet Connectivity between UOS and SOS:-

- Virtio driver pairs (FE & BE)
- Shared memory virtqueues between SOS & UOS: 1x Tx ring and 1x Rx ring (current design).
- UOS: standard Linux virtual NIC driver (discovered as PCI device)
- SOS: tap/tun driver + SW bridge +native Ethernet driver
- **Native Ethernet Driver owns the PTP clock.**

virtio framework lacks TSN support

Overview of Clock Source Technology in x86

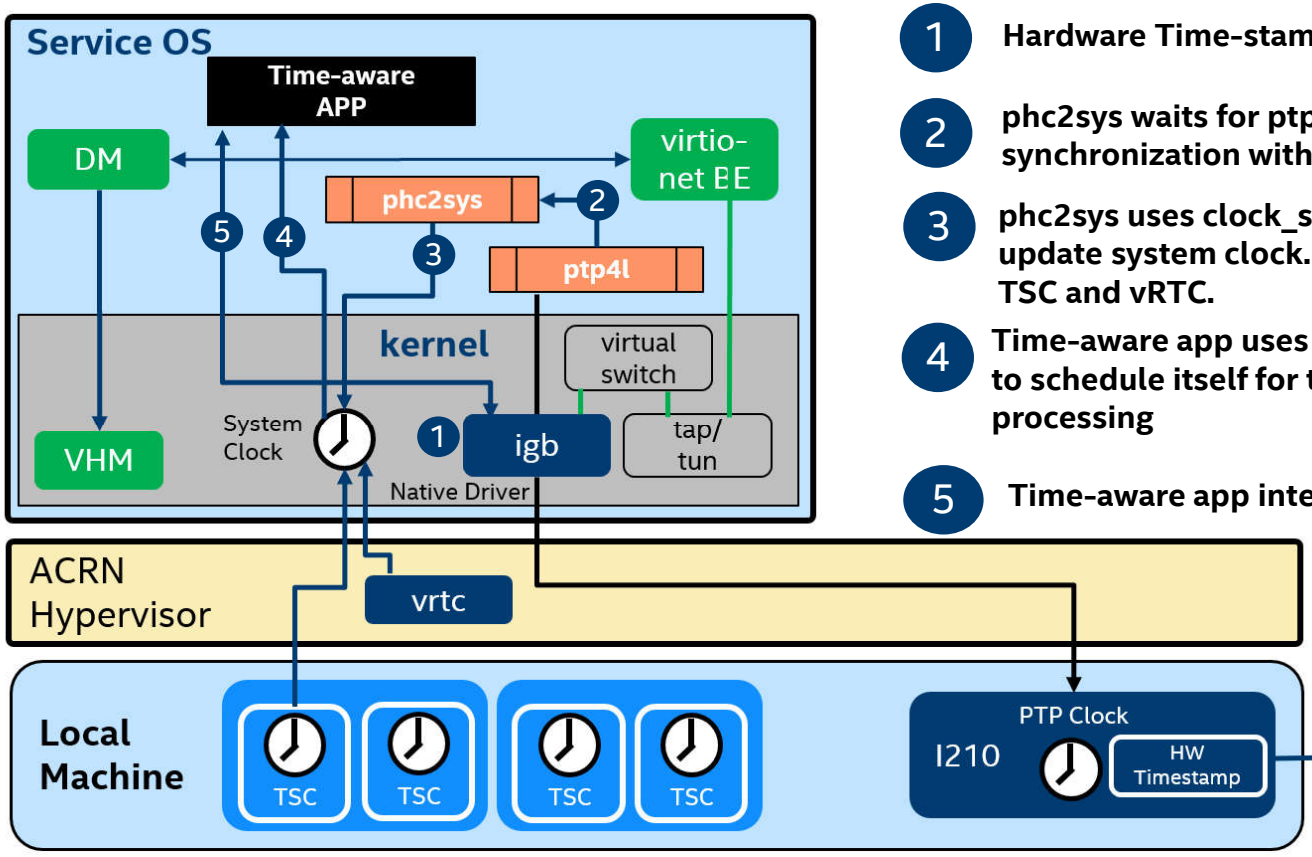
- Typical clock sources available for system clock:-
  - HPET (High Precision Event Timer) : 10MHz = one tick every 100ns
  - ACPI PM Timer : 3.58MHz = one tick every 279ns
  - **TSC (Time Stamp Counter) : Same as CPU frequency**
- **TSC** is commonly chosen due to high accuracy and low overhead.
- **Advancement in TSC** (check your CPU flags /proc/cpuinfo):-
  - **rdtscp :** atomic operation. 32-bit counter value & 32-bit auxiliary data (CPU ID)
  - **constant_tsc :** TSC counter synchronized across all sockets/cores.
  - **nonstop_tsc :** constant rate across PM states a.k.a. invariant TSC.

```
kernel $cat /proc/cpuinfo | grep tsc -m 1
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clfl
ush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx pdpe1gb rdtscp lm constant_tsc arch_per
fmon pebs bts rep_good nopl xtopology nonstop_tsc aperfmperf eagerfpu pni pclmulqdq dtes64 mon
itor ds_cpl vmx smx est tm2 ssse3 sdbg fma cx16 xtpr pdcm pcid dca sse4_1 sse4_2 x2apic movbe
popcnt tsc_deadline_timer aes xsave avx f16c rdrand lahf_lm abm epb tpr_shadow vnmi flexpriori
ty ept vpid fsgsbase tsc_adjust bmi1 avx2 smep bmi2 erms invpcid cqm xsaveopt cqm_llc cqm_occu
p_llc dtherm ida arat pln pts
```

Clocks in both Service OS & Guest OS

- In ACRN, both Service OS and Guest OS are virtual machines (VM).
- Currently, ACRN hypervisor fixed the CPU that VM runs on.
- ACRN VM depend on below sources to calculate its system clock:-
  - System-wide TSC (constant_tsc, nonstop_tsnc, rdtscp without VM Exit)
  - virtual RTC (vRTC) – Port I/O mediation for actual RTC device.
- RTC does not provide accurate real-time (temperature factor).
- To obtain more accurate real-time, use NTP or PTP.
- Under ACRN environment, Service OS has sole right to PTP hardware clock in Ethernet card. Its system time accuracy achieves PTP synchronization accurancy in nano-second range.
- Best to run time-sensitive application on Service OS.

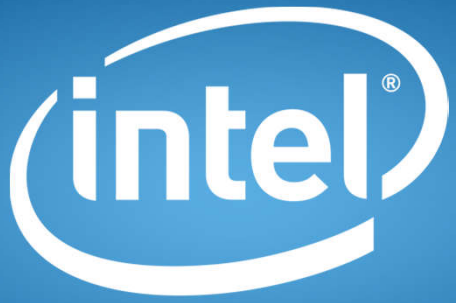
High Precision System Time in Service OS



1. Hardware Time-stamping used for PTP messages
2. phc2sys waits for ptp4l to enter locked state in PTP synchronization with master clock
3. phc2sys uses clock_settime(CLOCK_REALTIME) to update system clock. Note: system clock depends on TSC and vRTC.
4. Time-aware app uses clock_gettime(CLOCK_REALTIME) to schedule itself for time-sensitive workload processing
5. Time-aware app interacts with native Ethernet directly

Conclusion

- ACRN is FuSA certified, lightweight, real-time reference hypervisor and workloads (time-sensitive and not) can be consolidated on its VMs.
- Service OS has native Ethernet driver that solely owns PTP hardware clock.
- Hardware time-stamping achieves PTP clock synchronization accuracy in nano-second range. System clock time of SOS is synchronized to PTP clock.
- Virtio framework and virtual Ethernet do not have time-sensitive infrastructure.
- Therefore, for real-time workload, use SOS and direct the TSN traffics out of native Ethernet directly without going through virtual switch.



experience
what's inside™