Rethinking Timekeeping for Modern IT Solutions

Son VoBa, *Sync-n-Scale (VoBaS@sync-n-scale.com)*
Charles L. Ulland, *Sync-n-Scale (UllandC@sync-n-scale.com)*
Michael A. Lombardi, *NIST (Michael.Lombardi@nist.gov)*
Arno Lentfer, *The Windows Timestamp Project (al@windowstimestamp.com)*

ABSTRACT

Advances in processor performance and virtualization technologies, together with complexities of operating systems (OS) and software defined networking, have outpaced legacy network-based solutions for keeping system clocks accurate across geographic locations—in cloud-space and on-premises. Computer time accuracy with respect to Coordinated Universal Time (UTC) is now being codified and enforced by recently adopted regulatory requirements in the financial services sector for governance, risk mitigation, and demonstration of compliance.

These factors necessitate a rethinking of timekeeping for modern IT solutions. In this paper, we describe our technical approach to making UTC-accurate computer system clock a fundamental platform capability that is independent from other OS subsystems such as networking. We will also share a series of lessons learnt from building industry standard hardware interfaces to keep general-purpose computing platforms and their virtual machines UTC-accurate, and from our collaboration with industry partners Microsoft and selected server vendors to implement traceability to UTC via the National Institute of Standards and Technology (NIST) for modern IT solutions. We will conclude with our thoughts on how the full complement of Global Positioning System (GPS) positioning data in addition to accurate timekeeping would further benefit IT solutions of the future.

I. "REMEMBER THAT TIME IS MONEY" AND INACCURATE TIME CAN BE COSTLY

Benjamin Franklin was one of the founding fathers of the United States and a former Speaker of the Pennsylvania State House of Representatives. He was also an author, printer, political theorist, politician, freemason, postmaster, scientist, inventor, humorist, civic activist, statesman, and diplomat. In his 1748 essay titled *Advice to a Young Tradesman*, Franklin wrote *"Remember that time is money."* He went on to elaborate in plain terms what is now called opportunity cost in micro-economic theory.

In modern day financial markets, conducting financial services using IT resources operated with inaccurate time with respect to UTC will inevitably incur real financial costs and damaged business reputation. These costs come in the form of levied fines and publicly sanctioned findings imposed by market authorities on a per infraction basis. In the United States, the Financial Industry Regulatory Authority (FINRA) is a regulatory body charged with governing business between brokers, dealers and the investing public. A survey of its published disciplinary actions for the years 2017 and 2018 shows that FINRA imposed fines of more than $6.5M against 31 member financial firms of all sizes for failures to keep accurate business computer time. These failures include *"report the* [in]*correct time of trade"*, *"time stamping inaccuracies"*, *"report order event timestamps* [not] *in milliseconds when the firm's system captured time in milliseconds"*, *"document the* [in]*correct execution time in trade memoranda"*, *"*[in]*accurately record the order receipt time by its financial advisors"*, and other infractions of similar nature. The firm Dealerweb **[1]** paid the smallest fine of $12.5K for failure to report the correct trade execution time for transactions to Trade Reporting and Compliance Engine (TRACE®). The largest fines of almost $3M were levied against Deutsche Bank Securities **[2]** for three episodes of numerous infractions each in this area of regulatory compliance requirements.

https://doi.org/10.33012/2019.16764

Financial services is one of the sixteen critical infrastructure sectors **[3]** the U.S. Department of Homeland Security has deemed vital to national security. These assets, systems, and networks, whether physical or virtual, are considered so essential that their incapacitation or destruction would have a debilitating effect on national security, economic stability, public health, and safety. These critical infrastructure components have become interdependent across vast regions, crossing jurisdictional and national boundaries and time zones. In the pursuit of "smart" operations, the need for precise position and increased dependency on accurate time have become more crucial for certain sectors such as energy, healthcare and public health, communications, and information technology. Dependency on legacy methods and tools leaves some sectors vulnerable to cyberattacks **[4]** and disruption. These attack vectors include time-shifting, *i.e.* changing time on computer systems; or preventing systems from synchronizing their clocks over the network, *i.e.* denial of service. Either causes the attacked targets to operate with incorrect time with respect to UTC.

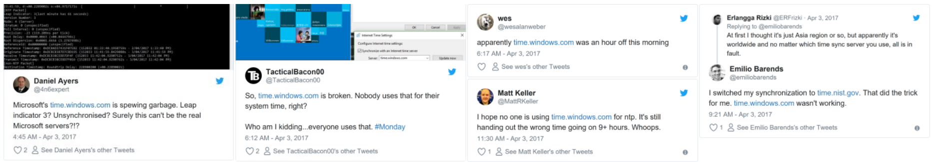
In its February 2015 Technical Note 1867 publication **[5]** the U.S. National Institute of Standards and Technology (NIST) asserted *"A new economy built on the massive growth of endpoints on the internet will require precise and verifiable timing in ways that current systems do not support."* NIST went on to elaborate *"Applications, computers, and communications systems have been developed with modules and layers that optimize data processing but degrade accurate timing."*

The relentless advances in processor performance and virtualization technologies continue unabated. These are being fully exploited by new features and abstractions in commercial mainstream operating systems and combined with innovative software defined networking capabilities to become a rich and flexible foundation for modern IT solutions and hybrid deployments—both in cloud-space and on-premises.

Against this fast-moving tide, legacy network-based time synchronization methods and tools remain stagnant. Their utility is often only suitable for specific IT scenarios and topologies where by design the underlying components are not subjected to constant operational changes, or on-demand layers of abstraction. Attempts at modernizing them are tepid or untenable in the face of more boundary conditions borne out of how IT is delivered as a service by the industry. In many cases, the fragile network time resources that are available are unreliable or lack accuracy measurement tools.

Once a carefully curated collection of industry resources, the NTP.org roster of stratum-one time servers lost its most ardent advocate when David L. Mills of the University of Delaware retired in 2008. Elsewhere on the internet, Ask Bjørn Hansen, the developer and maintainer of the crowd-sourced and volunteer-operated NTP Pool Project, put it *"If business, organization or human life depends on having correct time or can be harmed by it being wrong, you shouldn't "just get it off the internet"."*

Keeping computers UTC-accurate remains a persistent preoccupation of IT professionals, lest bad things happen. An April 2017 tweetstorm **[6]** (Fig. 1) illustrated this anxiety to be well founded and casts doubt on the accuracy and reliability of commercially-backed time services offered by a worldwide recognized IT vendor.



Daniel Ayers
@4n6expert
Microsoft's time.windows.com is spewing garbage. Leap indicator 3? Unsynchronised? Surely this can't be the real Microsoft servers?!?
4:45 AM - Apr 3, 2017
2 See Daniel Ayers's other Tweets

TacticalBacon00
@TacticalBacon00
So, time.windows.com is broken. Nobody uses that for their system time, right?

Who am I kidding...everyone uses that. #Monday
6:12 AM - Apr 3, 2017
2 See TacticalBacon00's other Tweets

wes
@wesalanweber
apparently time.windows.com was an hour off this morning
6:17 AM - Apr 3, 2017
See wes's other Tweets

Matt Keller
@MattRKeller
I hope no one is using time.windows.com for ntp. It's still handing out the wrong time going on 9+ hours. Whoops.
11:30 AM - Apr 3, 2017
1 See Matt Keller's other Tweets

Erlangga Rizki @ERFrizki · Apr 3, 2017
Replying to @emiliobarends
At first I thought it's just Asia region or so, but apparently it's worldwide and no matter which time sync server you use, all is in fault.

Emilio Barends
@emiliobarends
I switched my synchronization to time.nist.gov. That did the trick for me. time.windows.com wasn't working.
9:21 AM - Apr 3, 2017
1 See Emilio Barends's other Tweets

Figure 1. time.windows.com tweetstorm (April 3, 2017).

Other IT vendors adopted unconventional mechanisms to purposely alter the accuracy performance of their public facing network time services, *e.g.* leap-second smearing **[7]**. This is despite technical guidance to the contrary **[8]** from NIST.

A confluence of these emerging regulatory requirements, shared technical insights, and industry observations pointed to the need for rethinking timekeeping for modern IT solutions by:

- Introducing enterprise-grade computer hardware interfaces that exploit GPS signals as the primary and direct source for a disciplined oscillator to keep industry-standard general-purpose computer system clock UTC-accurate; and
- Collaborating with NIST and selected industry partners to devise methods and tools for definitive measurement of system clock accuracy, and for demonstration of such accuracy for regulatory compliance.

II. DESIGN APPROACHES AND FOUNDATIONAL TECHNOLOGIES

The Sync-n-Scale design approach for enabling UTC-accuracy to computer system clocks draws on the lessons learnt from the adoption of floating-point co-processor designs during the pioneering days of computing. That approach accelerated the computational speed while matching calculation precision delivered by the math software libraries that were commonly in use then. That approach also laid the foundation and charted a path for eventual hardware and software advances leading to the floating-point unit in modern day processor designs. Over the years, this same approach was taken by those in the audio video and cryptography fields.

A fundamental premise of this approach is that assured timing accuracy is an inherent computing platform capability and must persist as long as the platform is operating. An accurate system clock must not depend on auxiliary functions like networking interfaces and finicky external resources like “time servers” that are of dubious quality and trustworthiness.

This hardware-assisted approach does not impose any change requirements on user applications or deployed workloads. Nor does it require extraneous software tools added to the system. It draws on GPS time and frequency signals as a utility resource. Classic digital phase-locked loop methods are used to continually discipline the computer system clock through published kernel-mode APIs for assured persistent clock accuracy.

This approach augments the on-board time and frequency components. It transforms industry-standard servers and personal computers running mainstream commercially available operating systems into UTC-accurate general-purpose computing platforms suitable for all modern IT solutions.

The Sync-n-Scale PCIe small form-factor hardware interfaces are designed for cost-effective distribution of GPS time and frequency at data center scale. The GPS disciplined oscillator (GPSDO) interface (Fig. 2 left) connects a GPS antenna to the system. The expansion interface (Fig. 2 right) operates in tandem with the GPSDO to bring the same GPS time and frequency signals to the next system.

The Sync-n-Scale hardware interfaces allow one (Fig. 3 top and bottom left) or many computer systems (Fig. 3 right) to operate with a single GPS antenna in standard configuration. These systems and their virtual machines operate with UTC-accuracy in all environments, including those where networking connectivity is not present. The reduced attack surface would yield a higher degree of operational assurance required of modern IT solutions.

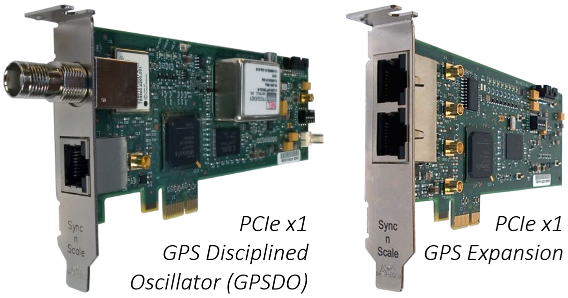


Figure 2. Sync-n-Scale PCIe hardware interfaces.

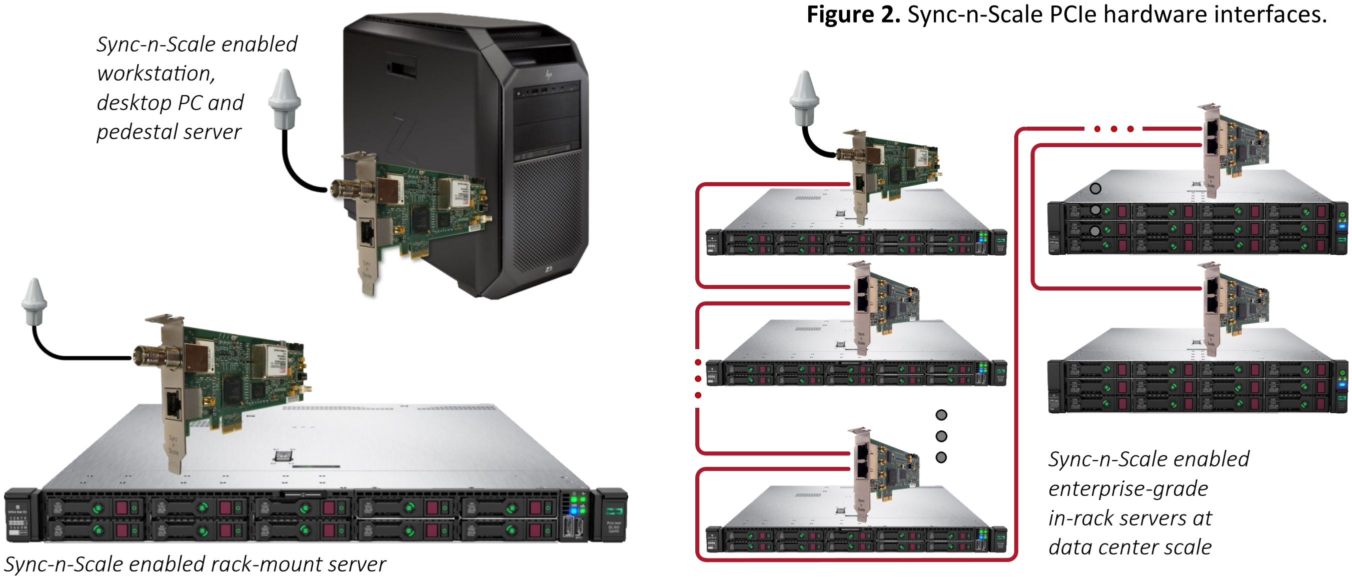


Figure 3. Sync-n-Scale enablement options for server and desktop systems.

The hardware interfaces are designed to also operate seamlessly in redundant configurations for assured-timing. The standard redundant configuration can be achieved with a starting pair of industry-standard servers of same or different makes and models (Fig. 4). Additional servers can then be added to the signal distribution topology (Fig. 5) with a redundant interface configuration (Fig. 5 bottom right) or standard configuration (Fig. 5 bottom left the HPE EdgeLine EL4000 chassis houses four individual system cartridges, each with its own PCIe slot).

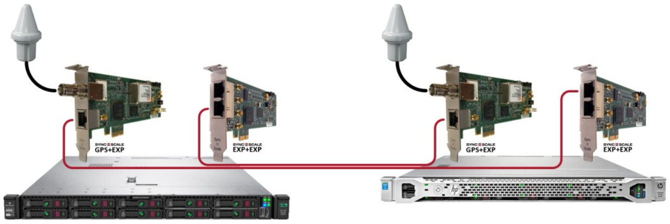


Figure 4. Redundant Sync-n-Scale enablement.

In these redundant configurations, there is one driver instance in each system. The driver designates one PCIe interface active and draws GPS time and frequency signals from it. When the currently active PCIe interface becomes inactive the driver automatically and seamlessly draws the signals from the other interface to continue operations.

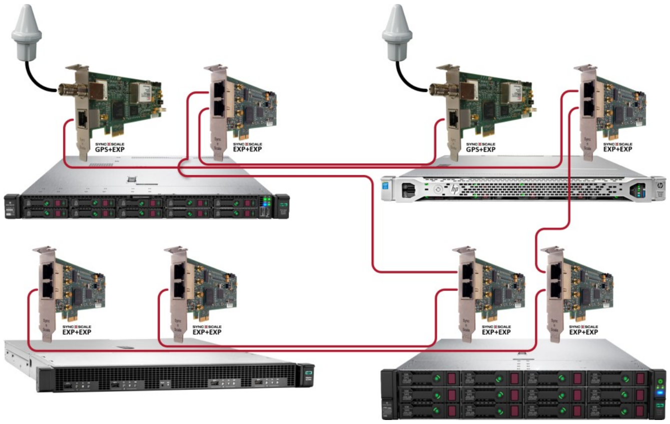
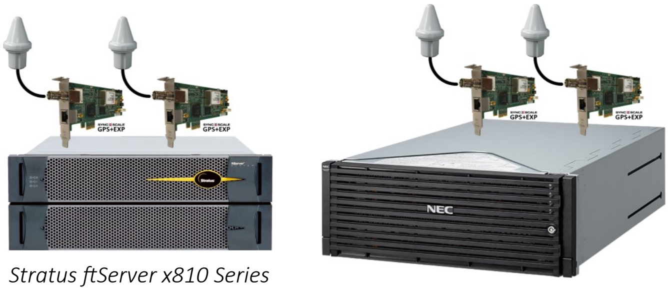


Figure 5. Redundant and Standard Sync-n-Scale enablement.

The above active-standby redundant configuration allows modern time-aware IT solutions to operate with high-availability and assured-timing on industry-standard x64 servers from a variety of vendors. For those that demand the continuous availability offered by fault-tolerant server platforms, the hardware interfaces are also a natural fit to keep these mission critical systems UTC-accurate (Fig. 6).



Stratus ftServer x810 Series

NEC Express5800/R320 Series

Figure 6. Sync-n-Scale enabled fault tolerant servers.

The fault-tolerant server architecture and lockstep hardware technology eliminate single points of failure and address the hardware, software, and serviceability issues that can lead to unplanned downtime and corruption or loss of critical data. Replicated hardware components process the same instructions at the same time. In the event of a component malfunction, processing does not miss a beat. If the system cannot automatically correct the problem, the part with the problem stops while the rest of the system continues normal operations without system downtime or data loss. This is accomplished without the need for failover scripting, or any extra effort to make applications cluster-aware **[9]**.

In a fault-tolerant server, a pair of PCIe interfaces are operated concurrently, each by its own device driver instance and connected to a separate GPS antenna in an active-active configuration. Measurements of 1 pulse per second (PPS) show the two interfaces diverge less than 50 ns from each other and from UTC(NIST) (Fig. 7).

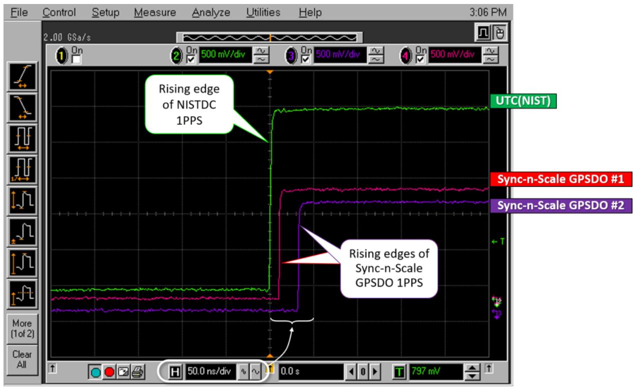


Figure 7. Scope screen capture.

III. FOUNDATIONAL TECHNOLOGIES

Hardware-assisted enablement requires bringing the same GPS time and frequency signals directly and efficiently to the general-purpose computing end points, or as close to their software abstraction equivalences as possible. This allows these end points to operate applications and workloads at UTC-accuracy.

The Synchronized Crystal Oscillator (SXO) and Oven Controlled Crystal Oscillator (OCXO) are foundational technologies making this approach possible at the data center scale. Other commercial-off-the-shelf (COTS) components are designed in, including a GPS receiver module and antenna, LMR-400 coaxial of up to 100 m **[10]** in length for antenna connectivity, Cat-5 cable (with RJ45 proprietary pin-out) for expansion connectivity, FPGA for device programmability, and PCIe x1 host interface.

The SXO's frequency and phase lock to an epoch, *e.g.* UTC(USNO) via GPS signals. The interconnected SXOs operate and behave as a single crystal oscillator. All points on the physical link of the interconnected SXOs are synchronous in frequency and do not have any noticeable dynamic phase error. Static phase error (which is constant for each connected SXO node) can be corrected as needed by other elements in an SXO designed-in device. Signals on the bus are cleaned up by the recursive filtering of each SXO unit. Phase noise (and jitter) on the bus signal is as good as the best SXO in the ensemble.

Failure of any arbitrary number of individual SXO does not lead to the system failure. All remaining SXO designed-in devices stay synchronous and provide signals on the buses to be tapped off by the system. The interconnected SXOs do not require a master clock to operate, therefore there is no single point of failure.

The following diagram illustrates the Sync-n-Scale PCIe GPSDO (Fig. 8 left) and expansion (Fig. 8 right) interconnecting their respective SXOs over a multi-gigabit transceiver operating at 1.25 GHz serial bit rates to distribute GPS time and frequency signals throughout the ensemble. These physical links do not transport networking data of any type.

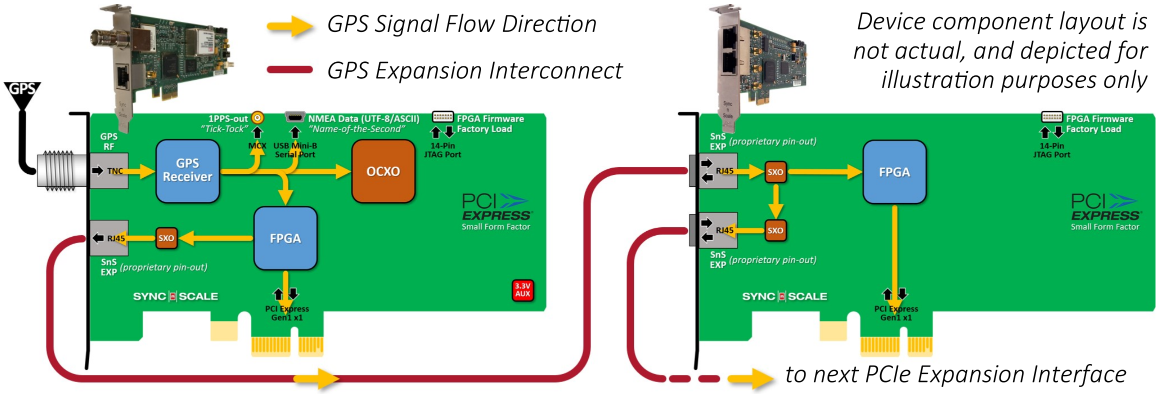


Figure 8. GPS signal flow through an ensemble of hardware interfaces.

The synchronicity performance of an interconnected ensemble of SXOs remains consistent across all instances regardless of their total count. By design, there is no architectural limit to the device count in an ensemble. The in-lab harness recorded this synchronicity precision performance at a constant 10 ns offset between the first and all other PCIe interfaces in the ensemble (Fig. 9).

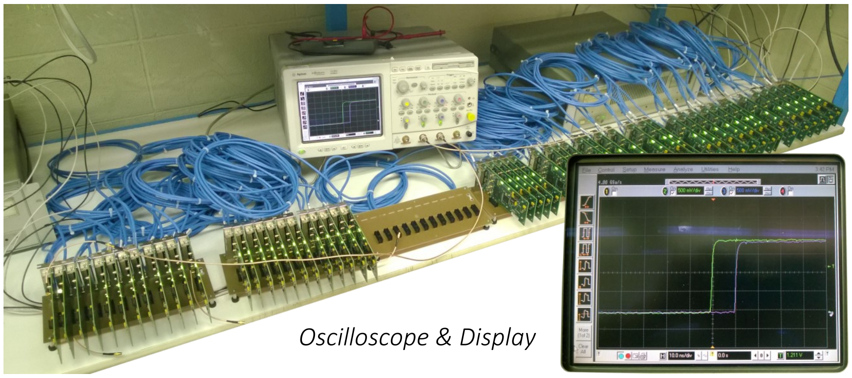


Figure 9. SXO precision measurement and harness.

The GPSDO PCIe interface contains an OCXO which operates at 62.5 MHz as the on-board backup frequency reference. Should the GPSDO detect GPS-denied conditions due to receiver hardware faults or external environmental factors, its FPGA firmware would take remedial steps. These include switching the 1 PPS source from the GPS receiver to the on-board OCXO, and alerting the system of this holdover transition, while maintaining the system clock accuracy in best effort indefinitely, or until GPS signals are successfully re-acquired.

The GPSDO OCXO holdover is rated for 8 µs over 8 hours (at ambient room temperature). Holdover measurements conducted at NIST-JILA laboratories demonstrated that a Windows Server 2016 system clock diverged less than 1 ms from UTC(NIST) in 21 days. Field units may vary.

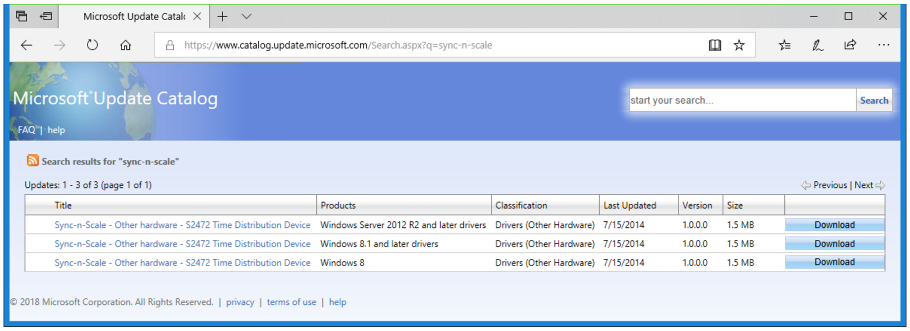
The OCXO "Europack" package footprint allows drop-in of better holdover performance and future design replacement parts during manufacturing of the Sync-n-Scale GPSDO interface.

IV. OPERATING SYSTEM INTERFACE

This section focuses on Sync-n-Scale enablement of Microsoft Windows desktop and server operating systems, and Hyper-V hypervisor as platforms for Windows and non-Windows apps and workloads requiring UTC-accurate timestamps.

In addition to the GPS hardware interfaces with their design-in specialty oscillators, Sync-n-Scale creates differentiation and adds values to these software stacks through development of its device firmware and enterprise-grade Windows kernel-mode device driver. The collaboration between Sync-n-Scale and Microsoft resulted in new capabilities in these software stacks for disciplining system clock, and improving accuracy of system clocks in physical systems, virtual machines, and other platform software abstractions such as containers.

The hardware interface is enabled and configured by Windows automatically or on demand through its Windows Update computer management tool and infrastructure (Fig. 10 right). This chain of custody ensures the integrity of the driver codes, and a zero-touch hardware installation experience. The device driver is also available for download from the Microsoft Update Catalog (Fig. 10 left) to accommodate disconnected deployments.



Search results for "sync-n-scale"

Updates: 1 - 3 of 3 (page 1 of 1)

Title	Products	Classification	Last Updated	Version	Size	
Sync-n-Scale - Other hardware - S2472 Time Distribution Device	Windows Server 2012 R2 and later drivers	Drivers (Other Hardware)	7/15/2014	1.0.0.0	1.5 MB	Download
Sync-n-Scale - Other hardware - S2472 Time Distribution Device	Windows 8.1 and later drivers	Drivers (Other Hardware)	7/15/2014	1.0.0.0	1.5 MB	Download
Sync-n-Scale - Other hardware - S2472 Time Distribution Device	Windows 8	Drivers (Other Hardware)	7/15/2014	1.0.0.0	1.5 MB	Download

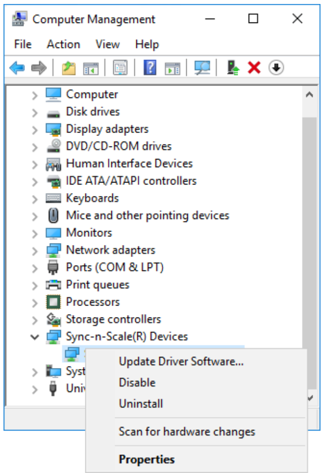
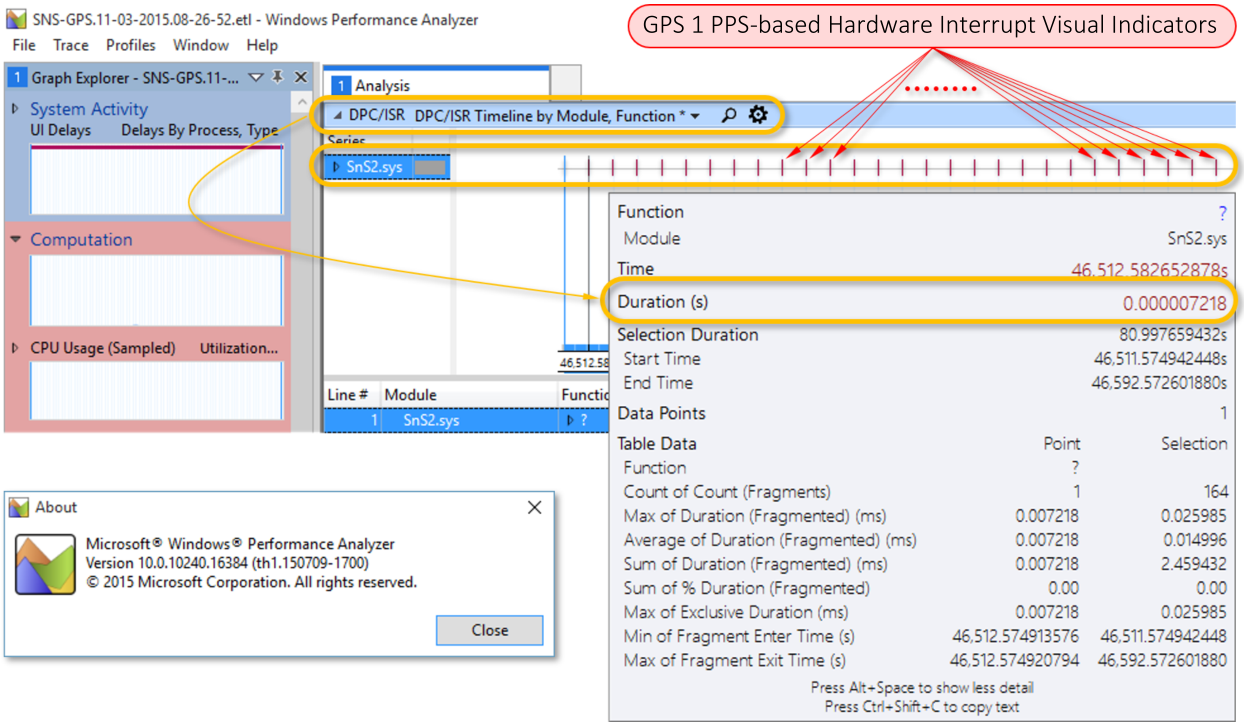


Figure 10. Sync-n-Scale kernel-mode driver download and installation.

In a 64-bit Windows OS on "bare-metal", the kernel-mode device driver and system service operate independently of the in-box Windows Time Services (W32Time) **[11]**. Specifically, they do not make use of the Windows Time Provider **[12]** plug-in model and framework for disciplining the system clock. This clean separation of maintaining a UTC-accurate general computing platform without consuming time as a service is necessary for modern IT solutions.

The interface generates a PCIe bus hardware interrupt announcing the arrival of each GPS 1 PPS signal. In response to these hardware interrupts, the kernel-mode driver and system service "advise" the Windows OS by how much to adjust the system's time-of-day clock. These "speed-up" and "slow-down" adjustments aim to coincide the next system clock second boundary with the next arrival of the GPS 1 PPS signal, keeping these ephemeral events as close to each other as possible. The hardware and software components do not affect changes to the system clock directly.

The PCIe hardware timing behaviors and interrupts, alongside driver and OS responses can be recorded, measured and inspected individually using the Windows Performance Recorder and Analyzer **[13]** tools (Fig. 11).



Microsoft® Windows® Performance Analyzer
Version 10.0.10240.16384 (th1.150709-1700)
© 2015 Microsoft Corporation. All rights reserved.

Function		?
Module		SnS2.sys
Time		46,512.582652878s
Duration (s)		0.000007218
Selection Duration		80.997659432s
Start Time		46,511.574942448s
End Time		46,592.572601880s
Data Points		1
Table Data	Point	Selection
Function	?	
Count of Count (Fragments)	1	164
Max of Duration (Fragmented) (ms)	0.007218	0.025985
Average of Duration (Fragmented) (ms)	0.007218	0.014996
Sum of Duration (Fragmented) (ms)	0.007218	2.459432
Sum of % Duration (Fragmented)	0.00	0.00
Max of Exclusive Duration (ms)	0.007218	0.025985
Min of Fragment Enter Time (s)	46,512.574913576	46,511.574942448
Max of Fragment Exit Time (s)	46,512.574920794	46,592.572601880

Press Alt+Space to show less detail
Press Ctrl+Shift+C to copy text

Figure 11. Windows Performance Analyzer display of Sync-n-Scale kernel-mode driver performance data.

These "steering" advisory operations could complete as fast as within 7 µs and on average within approximately 15 µs. They may be initiated more than once per second, or not at all, depending on the system time accuracy at that moment. As a result, the system clock rapidly becomes UTC-accurate shortly after Windows completes its kernel loading phase during boot or resume. More importantly, these operations also keep the system clock persistently accurate even when it is forced to a bogus time or date by a rogue system administrator **[14]** or infiltrating malware.

IV.1. VIRTUAL MACHINE OPERATING SYSTEMS

Public clouds such as Microsoft Azure Cloud offer on-demand computing resources through the internet on a paid subscription basis. These cloud-based resources include Hyper-V virtual machines (VM), and other computational services such as planet-scale distributed databases, directory services, crypto key repositories, *etc.* Sync-n-Scale and server vendor Hewlett Packard Enterprise collaborated with Microsoft to bring Sync-n-Scale enabled HPE ProLiant for Microsoft Azure Stack **[15]** (HPE AzS) to customer on-premises data centers and private clouds. This approach to IT infrastructures allows organizations to take advantage of the rapid innovation and agility of cloud computing, and to meet regulatory requirements of their industries.

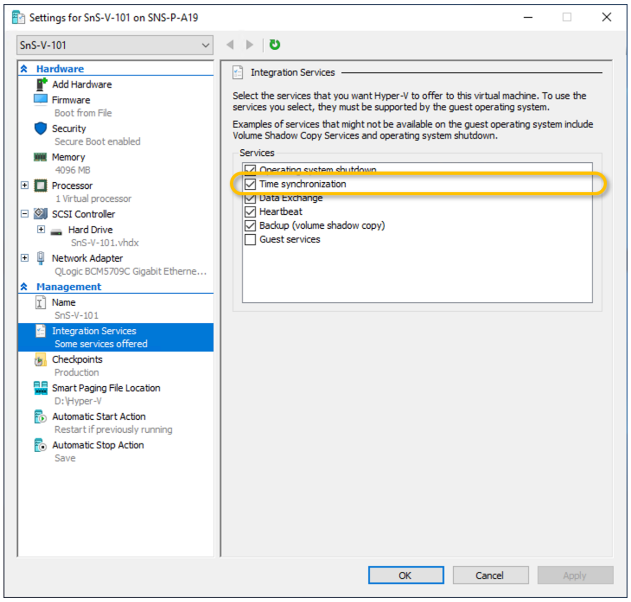
Certain timestamp accuracy requirements demanding stringent evidence-based performance **[16]** that could only be met by cost-effective infrastructure like Sync-n-Scale enabled HPE AzS. Collectively, the hybrid-IT **[17]** model (cloud-based and on-premises) and approach make it flexible for organizations to implement and deploy modern IT solutions. In addition to Microsoft Azure Cloud and its Azure Stack partners, Amazon Web Services and its Outposts partners have also started pursuing similar hybrid IT platform strategies **[18]**.

Microsoft Hyper-V VMs enhance their performance by using the VMBus **[19]**, which is a two-way in-memory channel-based mechanism, to communicate with their Hyper-V host. The collection of Hyper-V Integration Services **[20]** in the host and VMs consists of Windows system services and drivers. They are called Virtual Machine Integration Components (VMIC or "VM IC"). The comparable ones for supported Linux and FreeBSD **[21]** VMs are called Linux Integration Services (LIS) **[22]**.

Most Windows VMs are set up to get important updates from Windows Update, thus these integration services are kept up to date by default in order to deliver the best performance and most recent features. Linux and FreeBSD VMs receive the latest integration components when the VM kernel is updated by the system administrator.

Many of these services are conveniences such as guest file copy, while others such as synthetic device drivers that are important to the virtual machine's functionality, and high-impact services such as time synchronization. For any given virtual machine, individual convenience services can be enabled to operate or not; while others are not intended to be serviced manually. Non-Windows and earlier releases of Windows guest operating systems might not have all available services. To work correctly, each desired integration service must be enabled on both the host and guest.

The Hyper-V Time synchronization **[23]** service (Fig. 12) retrieves time from the Hyper-V physical host and forwards it on to its counterpart in the VM. If this high-impact service is disabled on the Hyper-V host side, the VM system clocks will drift erratically.



Settings for SnS-V-101 on SNS-P-A19

SnS-V-101

Hardware

Add Hardware

Firmware
Boot from File

Security
Secure Boot enabled

Memory
4096 MB

Processor
1 Virtual processor

SCSI Controller
Hard Drive
SnS-V-101.vhdx

Network Adapter
QLogic BCM5709C Gigabit Etherne...

Management

Name
SnS-V-101

Integration Services
Some services offered

Checkpoints
Production

Smart Paging File Location
D:\Hyper-V

Automatic Start Action
Restart if previously running

Automatic Stop Action
Save

Integration Services

Select the services that you want Hyper-V to offer to this virtual machine. To use the services you select, they must be supported by the guest operating system.

Examples of services that might not be available on the guest operating system include Volume Shadow Copy Services and operating system shutdown.

Services

☑ Operating system shutdown
☑ Time synchronization
☑ Data Exchange
☑ Heartbeat
☑ Backup (volume shadow copy)
☐ Guest services

Figure 12. Hyper-V Virtual Machine Manager settings.

Inside a Windows VM, the VM Time Provider (VMTP) is an active component of W32Time. The Windows VM acts as an NTP client by relying on the VMTP to synchronize its system clock with that of the Hyper-V host. The Hyper-V host acts as an NTP server at a “higher” stratum. In NTP principles, a Sync-n-Scale enabled Hyper-V host occupies stratum 1, and its VMs are all at stratum 2.

The Linux VM VMICTimeSync provider implements **hyperv** as a Precision Time Protocol (PTP) clock source **[24]** for synchronizing its system clock with that of the Hyper-V host. The Hyper-V host acts the role of a PTP grandmaster providing synchronization to multiple VM clients sharing the same VMBus network.

Regardless of the time synchronization methods and implementations, these in-memory mechanisms let the VMs take a dependency on the Hyper-V host for time synchronization with the least uncertainty factors in calculating the clock offset between the two systems. This assertion is demonstrably achievable without being heroic when the Sync-n-Scale enabled Hyper-V host time accuracy is a control variable. Conversely, if the Hyper-V host time accuracy cannot be kept persistent for any particular reason, the time accuracy of all of its VMs “suffers” accordingly.

IV.2. WORKING WITH WINDOWS CONTAINER IMAGES

Container is an emerging platform technology for modern IT solutions and is uneven in areas that might affect time-aware applications. The Microsoft base container images **[25]** are available with either Windows Server Core or Nano Server as the container operating system.

At this time, Nano Server base container images do not provide the full complement of time and date capabilities and tools. For modern IT solutions, time-aware Windows applications and scripts would best operate in a Windows Server Core base container image. The Windows Time service (W32Time) requires a manual start in the Microsoft base container image.

The CMD commands **time** and **date** are built-in commands. Because they are not separate executables in the Windows Container images, they need to be run in an interactive **cmd** shell, or executed, *e.g.* **docker exec –it** <*container-image*> **cmd /c time /T**. The **cmd** variables **%time%** and **%date%** are also available, *e.g.* **docker exec –it** <*container-image*> **cmd /c echo %time%**.

Given the temporary and transient nature of containers by design, architects and developers of applications that are required to demonstrate time accuracy (*i.e.* traceability) may need to explore operational and deployment methods to persist the identity of their container instance for historical references.

V. TOOLINGS FOR MONITORING AND MEASUREMENT

Modern regulatory requirements for business computer time accuracy in the financial services industry have evolved to include more stringent outcome-based performance and evidence-based governance. Business decision makers and stake holders look to their IT organizations for time accuracy performance of deployed IT solutions as an on-going governance metric to avoid regulatory enforcement consequences for not meeting these requirements.

Expertise in time metrology is not a priority for the IT community and its professionals. The scenarios they need to address might be expressed as follows:

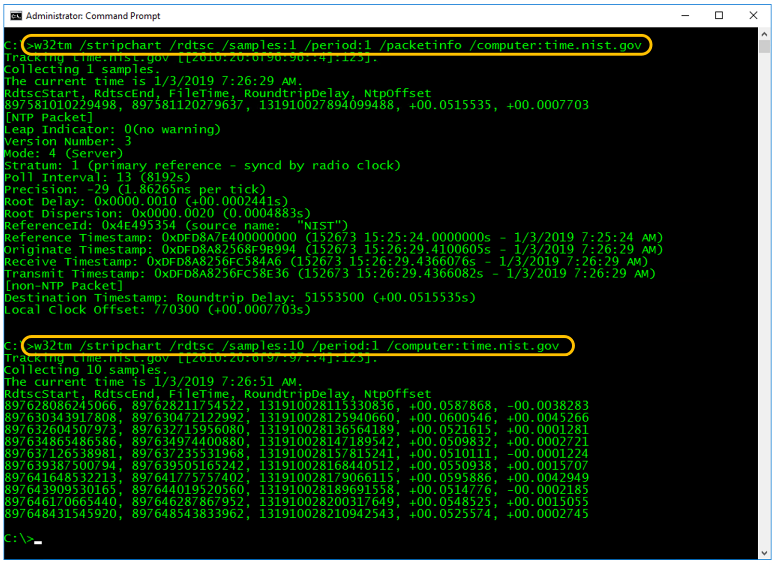
1. At any moment, how does one know if a system clock is accurate and within a prescribed precision range of UTC?
2. In a legal proceeding, how does one prove a system clock was UTC-accurate at any given point in the past?

This section describes selected readily available tools and reproducible harnesses for IT professionals to demonstrate business computer UTC-accuracy to themselves, their users and other stake holders.

V.1. MICROSOFT WINDOWS IN-BOX AND OPEN-SOURCE TOOLS

Microsoft made engineering investments to improve time accuracy of Windows desktop and server operating systems, and Hyper-V hypervisor. Given the legacy of Windows, it is best that IT professionals filter out older on-line references, community “guidance,” and commentaries pre-dating the Windows Server 2016 and Windows 8.1 releases. Microsoft developers also made available open source tools **[26]** that are deemed useful for measurement and characterization of system time accuracy.

w32tm.exe is the in-box preferred tool for configuring, monitoring and diagnosing the Windows Time service (W32Time). Its Time Stamp Counter (TSC) read can be used for time accuracy measurement of Sync-n-Scale enabled Hyper-V Windows VMs and Windows Containers (Fig. 13).



```
C:\>w32tm /stripchart /rdtsc /samples:1 /period:1 /packetinfo /computer:time.nist.gov
Tracking time.nist.gov [[2610:20:6f96:96::4]:123].
Collecting 1 samples.
The current time is 1/3/2019 7:26:29 AM.
RdtscStart, RdtscEnd, FileTime, RoundtripDelay, NtpOffset
897581010229498, 897581120279637, 131910027894099488, +00.0515535, +00.0007703
[NTP Packet]
Leap Indicator: 0(no warning)
Version Number: 3
Mode: 4 (Server)
Stratum: 1 (primary reference - syncd by radio clock)
Poll Interval: 13 (8192s)
Precision: -29 (1.86265ns per tick)
Root Delay: 0x0000.0010 (+00.0002441s)
Root Dispersion: 0x0000.0020 (0.0004883s)
ReferenceId: 0x4E495354 (source name:  "NIST")
Reference Timestamp: 0xDFD8A7E400000000 (152673 15:25:24.0000000s - 1/3/2019 7:25:24 AM)
Originate Timestamp: 0xDFD8A82568F9B994 (152673 15:26:29.4100605s - 1/3/2019 7:26:29 AM)
Receive Timestamp: 0xDFD8A8256FC584A6 (152673 15:26:29.4366076s - 1/3/2019 7:26:29 AM)
Transmit Timestamp: 0xDFD8A8256FC58E36 (152673 15:26:29.4366082s - 1/3/2019 7:26:29 AM)
[non-NTP Packet]
Destination Timestamp: Roundtrip Delay: 51553500 (+00.0515535s)
Local Clock Offset: 770300 (+00.0007703s)


C:\>w32tm /stripchart /rdtsc /samples:10 /period:1 /computer:time.nist.gov
Tracking time.nist.gov [[2610:20:6f97:97::4]:123].
Collecting 10 samples.
The current time is 1/3/2019 7:26:51 AM.
RdtscStart, RdtscEnd, FileTime, RoundtripDelay, NtpOffset
897628086245066, 897628211754522, 131910028115330836, +00.0587868, -00.0038283
897630343917808, 897630472122992, 131910028125940660, +00.0600546, +00.0045266
897632604507973, 897632715956080, 131910028136564189, +00.0521615, +00.0001281
897634865486586, 897634974400880, 131910028147189542, +00.0509832, +00.0002721
897637126538981, 897637235531968, 131910028157815241, +00.0510111, -00.0001224
897639387500794, 897639505165242, 131910028168440512, +00.0550938, +00.0015707
897641648532213, 897641775757402, 131910028179066115, +00.0595886, +00.0042949
897643909530165, 897644019520560, 131910028189691558, +00.0514776, -00.0002185
897646170665440, 897646287867952, 131910028200317649, +00.0548525, +00.0015055
897648431545920, 897648543833962, 131910028210942543, +00.0525574, +00.0002745

C:\>_
```

Figure 13. Microsoft Windows **w32tm /stripchart /rdtsc** command.

The TSC is a 64-bit register present on all x86/64 processors since the Intel Pentium generation. It counts the number of cycles since last reset. Invariant TSC can be found in modern processor designs and operates at a constant rate regardless of the processor operating frequency and power saving states. Both host and guest can query the common processor TSC using the instruction RDTSC. TSC reads are efficient and do not incur the overhead associated with a ring transition or access to a platform resource. When queried in short durations (*e.g.* one second apart for less than 10 minutes) the TSC values are very accurate. The command **w32tm /stripchart /rdtsc** is to be run simultaneously in both the Hyper-V host and the calibrated Windows VM guest. In each system, the tool records the system's time-of-day timestamps and corresponding TSC values. Because these values are retrieved from a common invariant TSC, they can then be used to calculate the clock offsets between the two systems directly.

For those who are inclined towards replicating the measurement harness for a mix OS environment of hypervisor host and VM guest platforms, the Microsoft tool OS time sampler **[27]** is a good starting point. It is limited to recording the local system time-of-day timestamps and corresponding invariant TSC values. The command **ostimesampler 1000 500** takes 500 samples, each 1 second apart. Additional capabilities can be added to the original source codes as appropriate.

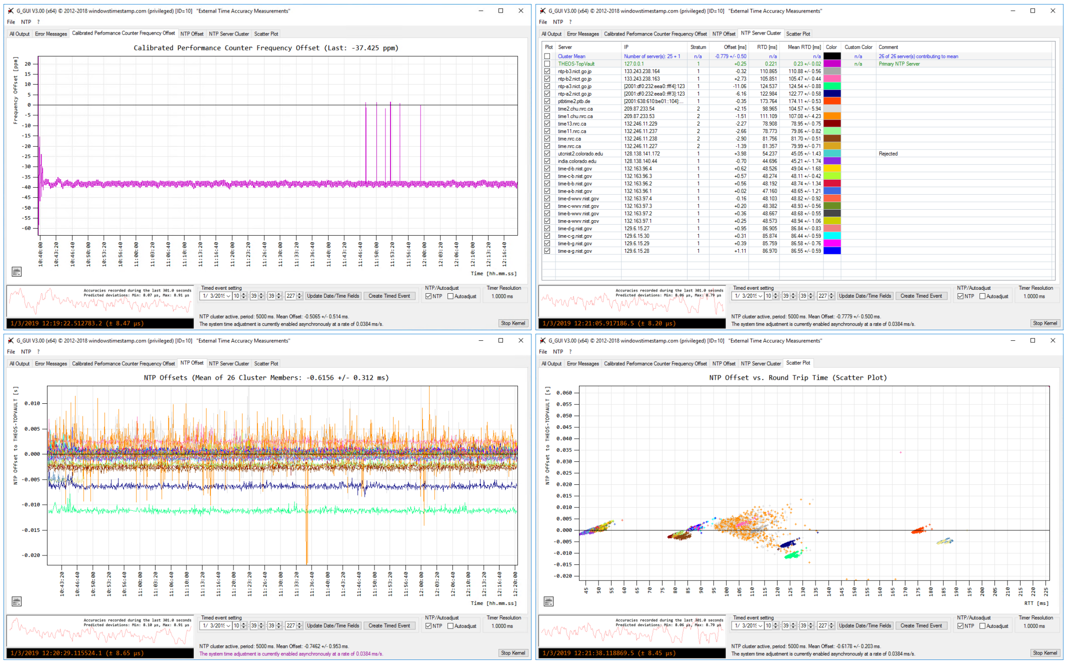
These command line tools are best for a harness that allows them to be run in both hypervisor host and its VM guest platforms. Doing so minimizes or altogether eliminates the contributing uncertainty factors that would skew the clock offset calculation results. Hence, these tools are not practical for evaluating system time accuracy of public cloud VMs such as those in Azure Cloud because access to the underlying Hyper-V hypervisor is not possible for the general population.

Instead, comparable on-premises alternative such as the HPE AzS would be more suitable for these tools to ascertain VM time accuracy performance. In-house IT development resources, or qualified outside service organizations such as HPE Pointnext can also be called upon to create sophisticate harness using these tools for complex enterprise needs.

These command line tools generate numeric data that can be consumed by other tools. This style of analytic workflow is suboptimal for real-time and long-haul visualization of a system time accuracy performance and trend. Furthermore, such workflow would be cumbersome at scale for multiple monitored and measured systems.

V.2. THE WINDOWS TIMESTAMP PROJECT TOOL SUITE

The Windows Timestamp Project G Suite **[28]** is an independent and free tool suite. It is a ready-made expert alternative to the Microsoft in-box command line tools. The G Suite GUI includes visualizing stability of the local system clock over time (Fig. 14 top left), the monitoring of up to 64 NTP time sources (Fig. 14 top right), visualizing their clock synchronicity with



Plot	Server	IP	Stratum	Offset [ms]	RTD [ms]	Mean RTD [ms]	Color	Custom Color	Comment
☐	Cluster Mean	Number of server(s): 25 + 1	n/a	-0.779 +/- 0.50	n/a	n/a	■	n/a	26 of 26 server(s) contributing to mean
☐	THEOS-TopVault	127.0.0.1	1	+0.25	0.221	0.23 +/- 0.02	■	n/a	Primary NTP Server
☑	ntp-b3.nict.go.jp	133.243.238.164	1	-0.32	110.865	110.88 +/- 0.56	■		
☑	ntp-b2.nict.go.jp	133.243.238.163	1	+2.73	105.851	105.47 +/- 0.44	■		
☑	ntp-a3.nict.go.jp	[2001:df0:232:eea0::fff4]:123	1	-11.06	124.537	124.54 +/- 0.88	■		
☑	ntp-a2.nict.go.jp	[2001:df0:232:eea0::fff3]:123	1	-6.16	122.984	122.77 +/- 0.58	■		
☑	ptbtime2.ptb.de	[2001:638:610:be01::104]:...	1	-0.35	173.764	174.11 +/- 0.53	■		
☑	time2.chu.nrc.ca	209.87.233.54	2	+2.15	98.965	104.57 +/- 5.94	■		
☑	time1.chu.nrc.ca	209.87.233.53	2	-1.51	111.109	107.08 +/- 4.23	■		
☑	time13.nrc.ca	132.246.11.229	2	-2.27	78.908	78.95 +/- 0.75	■		
☑	time11.nrc.ca	132.246.11.237	2	-2.66	78.773	79.86 +/- 0.82	■		
☑	time.nrc.ca	132.246.11.238	2	-2.90	81.756	81.70 +/- 0.51	■		
☑	time.nrc.ca	132.246.11.227	2	-1.39	81.357	79.99 +/- 0.71	■		
☑	utcnist2.colorado.edu	128.138.141.172	1	+3.98	54.237	45.05 +/- 1.43	■		Rejected
☑	india.colorado.edu	128.138.140.44	1	-0.70	44.696	45.21 +/- 1.74	■		
☑	time-d-b.nist.gov	132.163.96.4	1	+0.62	48.526	49.04 +/- 1.68	■		
☑	time-c-b.nist.gov	132.163.96.3	1	+0.57	48.274	48.11 +/- 0.42	■		
☑	time-b-b.nist.gov	132.163.96.2	1	+0.56	48.192	48.74 +/- 1.34	■		
☑	time-a-b.nist.gov	132.163.96.1	1	+0.02	47.160	48.65 +/- 1.21	■		
☑	time-d-wwv.nist.gov	132.163.97.4	1	-0.16	48.103	48.82 +/- 0.92	■		
☑	time-c-wwv.nist.gov	132.163.97.3	1	+0.20	48.382	48.93 +/- 0.56	■		
☑	time-b-wwv.nist.gov	132.163.97.2	1	+0.36	48.667	48.68 +/- 0.55	■		
☑	time-a-wwv.nist.gov	132.163.97.1	1	+0.25	48.573	48.94 +/- 1.06	■		
☑	time-d-g.nist.gov	129.6.15.27	1	+0.95	86.905	86.84 +/- 0.83	■		
☑	time-c-g.nist.gov	129.6.15.30	1	+0.31	85.874	86.44 +/- 0.59	■		
☑	time-b-g.nist.gov	129.6.15.29	1	+0.39	85.759	86.58 +/- 0.76	■		
☑	time-a-g.nist.gov	129.6.15.28	1	+1.11	86.970	86.55 +/- 0.59	■		

```
Accuracies recorded during the last 301.0 seconds
Predicted deviations: Min: 8.07 µs, Max: 8.91 µs
1/3/2019 12:19:22.512783.2 (± 8.47 µs)
```

NTP cluster active, period: 5000 ms. Mean Offset: -0.5065 +/- 0.514 ms.
The system time adjustment is currently enabled asynchronously at a rate of 0.0384 ms/s.

```
Accuracies recorded during the last 301.0 seconds
Predicted deviations: Min: 8.06 µs, Max: 8.79 µs
1/3/2019 12:21:05.917186.5 (± 8.20 µs)
```

NTP cluster active, period: 5000 ms. Mean Offset: -0.7779 +/- 0.500 ms.
The system time adjustment is currently enabled asynchronously at a rate of 0.0384 ms/s.

```
Accuracies recorded during the last 301.0 seconds
Predicted deviations: Min: 8.10 µs, Max: 8.91 µs
1/3/2019 12:20:29.115524.1 (± 8.65 µs)
```

NTP cluster active, period: 5000 ms. Mean Offset: -0.7462 +/- 0.953 ms.
The system time adjustment is currently enabled asynchronously at a rate of 0.0384 ms/s.

```
Accuracies recorded during the last 301.0 seconds
Predicted deviations: Min: 8.06 µs, Max: 8.79 µs
1/3/2019 12:21:38.118869.5 (± 8.45 µs)
```

NTP cluster active, period: 5000 ms. Mean Offset: -0.6178 +/- 0.203 ms.
The system time adjustment is currently enabled asynchronously at a rate of 0.0384 ms/s.

Figure 14. The Windows Timestamp Project G Suite GUI display tabs.

respect to the local system time-of-day (Fig. 14 bottom left), and visualizing the effects of contributing uncertainty factors on round-trip delays (Fig. 14 bottom right). Visualized data can also be exported for further processing by other data tools.

The G Suite GUI helps visualizing the UTC-accuracy performance of a fault-tolerant server during its failover, and of a VM during its live migration across Hyper-V hosts, and comparable migratory operations offered by other hypervisors. The annotated chart (Fig. 15) was recorded inside a Windows Server 2016 VM being live migrated across two Sync-n-Scale enabled Hyper-V hosts. The close-up snippet of recorded data shows two clock offset "spikes". Each coincided with the start of a VM continued operations on the landing hypervisor immediately after the completion of its live migration. These clock offset spikes lasted about 90 seconds before the VM clock UTC-accuracy is returned to normal.

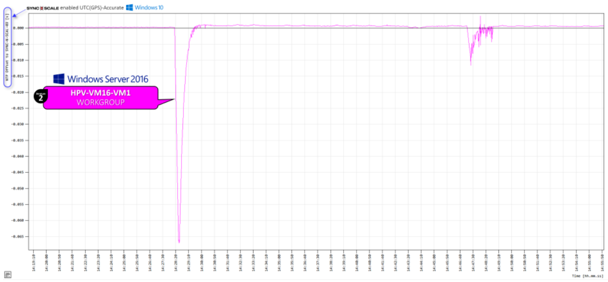


Figure 15. Hyper-V live migrated VM clock offset "spikes".

Another application of this tool is for understanding the underlying behaviors of the hypervisors. The following measurements were conducted inside the VMs being operated by cloud vendors AWS Xen (Fig. 16 left) and Azure Hyper-V (Fig. 16 right). The recorded clock offsets show how these hypervisors affect the VM time accuracy differently despite both were operating the same Windows Server 2012 R2 release and configured according to the vendors' prescription.

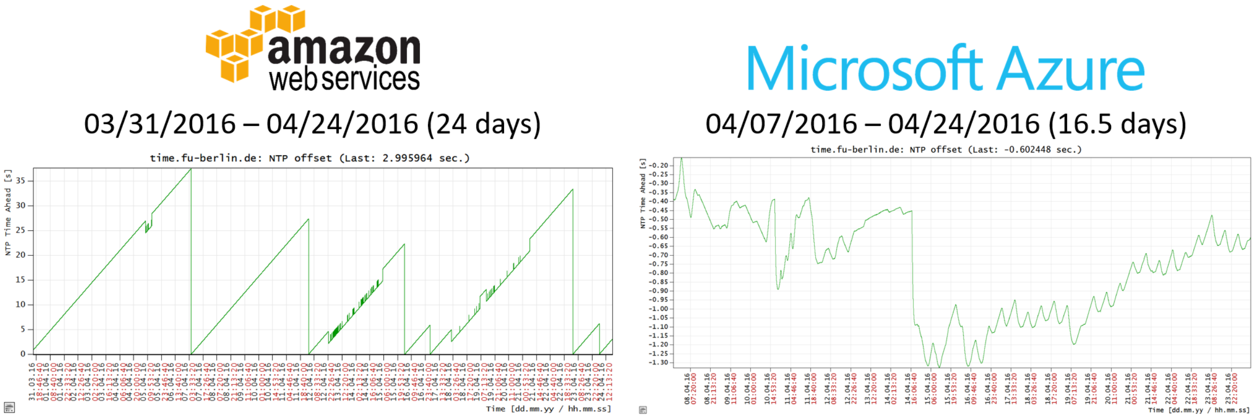


Figure 16. AWS and Azure VM time accuracy charts.

A more recent measurement was taken inside an Azure Cloud Windows Server 2016 VM which had been configured according to the recent blog Time sync for Windows VM in Azure **[29]**. The annotated chart (Fig. 17) shows a properly configured Azure Windows Server VM would improve its UTC-accuracy to fall within the FINRA/SEC 50 ms accuracy requirement **[30]**.

Availability of tools and consistency of harnesses shorten time to IT operational insights. Mainstream system management tools and dashboards still need to effectively integrate UTC-accuracy of IT resources as a meaningful performance metric in the overall health monitoring

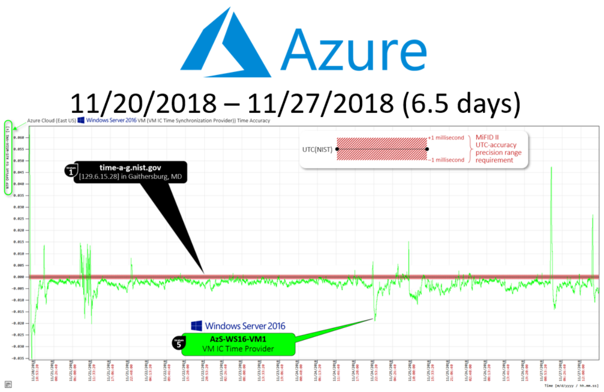


Figure 17. Azure VM time accuracy chart.

capabilities of modern IT solutions. As deployments of modern IT solutions are becoming complex across on-premises and cloud-based environments, outputs from these tools not only establish an informed basis for shared understanding of time accuracy performance, they become valuable input for machine learning and AI-based operational decisions.

Nevertheless, on their own and being operated by IT professionals these tools would not be adequate without corroborating UTC-accuracy measurements.

VI. THE ROLE OF NATIONAL LABORATORIES

At this point, we have described the solution building blocks for ensuring UTC-accuracy in modern IT solutions. Regulatory requirements make it imperative for IT professionals to demonstrate that their business computers are UTC-accurate. Effective corroboration of system time accuracy requires actual measurements using well-tested methods, tools and harnesses. An assertion of system time accuracy based on assumption, inference, or deduction should be received with a healthy dose of skepticism, regardless of its origin and no matter how well intended.

The U.S. National Institute of Standards and Technology (NIST) provides official time to the United States for commerce. Elsewhere, national metrology institutes (NMIs) **[31]** or designated time and frequency metrology laboratories who contribute to UTC serve as the reference timing source for their respective country. UTC(NIST) is the US coordinated universal time scale and the national standard for frequency and time interval. It comprises an ensemble of cesium and hydrogen maser atomic clocks that are maintained at the NIST laboratories in Colorado.

The public-private partnership with NIST has played a role in shaping our rethinking of timekeeping for modern IT solutions. Sync-n-Scale entered into a cooperative research and development agreement (CRADA) with the NIST Time and Frequency Division in April 2015 **[32]**. There were no monetary exchanges in this partnership. The initial objective for this collaboration was the *"Development of a Modern Framework, Associated Methods and Tools to Calibrate, Verify and Measure System Clocks of Independently Operated Servers and Personal Computers."* Since then, it has expanded to include IT centric metrology practices and modern instrumentation. This section describes the relevant collaboration output of this joint effort.

VI.1. NIST DISCIPLINED CLOCK (NISTDC)

Intuitively, and in an ideal world, it would make sense to place an in-production computer physically next to a national time scale, and to utilize the full complement of NMI tools, harnesses, and subject matter experts to definitively measure its system clock divergence from UTC. Alas, it is quickly obvious that approach would not be at all practical and scalable in the IT world.

It was from that learning, that an alternative was sought. The NIST Disciplined Clock (NISTDC) is an available offering from NIST as an optional add-on to its Time Measurement and Analysis Service (TMAS).

The NISTDC is an active monitoring and measurement instrument for modern IT solutions (Fig. 18). Each NISTDC is a replica of the US time scale for time interval (second) and frequency (hertz) metrology purposes. It is highly calibrated and contains no self-serviceable hardware and software parts.

Figure 18. Sync-n-Scale operated NISTDC unit.

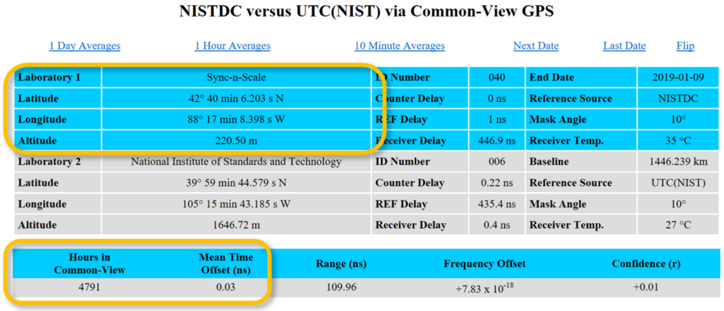
The NISTDC operates as an internet connected device with its own GPS antenna and calibrated coaxial cable. It uses both the internet, and common-view and all-in-view **[33]** observations of GPS satellites, and is directly referenced to the UTC (NIST) time scale in Boulder, Colorado. These capabilities allow the NISTDC to be deployed and functional worldwide.

The NISTDC is built on a BitLocker-protected **[34]** 64-bit Windows 10 platform. The NIST instrumentation and calibration software disciplines an in-chassis rubidium clock to agree with UTC(NIST). This makes it possible to definitively establish UTC-traceability of the monitored and measured systems to the International System (SI) Units Time **[35]** directly through NIST.

The NISTDC frequency uncertainty is rated at less than 1 × 10^{-14} after one day of averaging with respect to UTC(NIST), and time uncertainty is 12 to 15 ns with respect to UTC(NIST). These NISTDC performance metrics and its actual geolocation are continuously recorded in the NIST data repository and can be viewed using a standard web browser.

The NISTDC can be deployed inside the IT perimeter of an organization, and within intranet proximity of its monitored and measured in-production computer systems, while operating outside of the management domains of those systems. It does not analyze or graph time accuracy data. It collects and stores the measurement data locally, before forwarding them over the internet to the NIST repository for archival. UTC(NIST)-accuracy analysis of the monitored and measured systems can then be securely viewed in tabular and graphical form using a standard web browser.

A Sync-n-Scale operated NISTDC had been calibrated at NIST in Boulder, Colorado prior to shipment to its current location in Burlington, Wisconsin. The configured NIST software tools continue this calibration on-site and the accuracy of the UTC(NIST) replica can be examined in real time. An RMS 0.03 ns divergence from UTC(NIST) is nominal in this 200-day measurement ending on January 9, 2019 (Fig. 19).



NISTDC versus UTC(NIST) via Common-View GPS

1 Day Averages 1 Hour Averages 10 Minute Averages Next Date Last Date Flip

Laboratory 1	Sync-n-Scale	**ID Number**	040	**End Date**	2019-01-09
Latitude	42° 40 min 6.203 s N	**Counter Delay**	0 ns	**Reference Source**	NISTDC
Longitude	88° 17 min 8.398 s W	**REF Delay**	1 ns	**Mask Angle**	10°
Altitude	220.50 m	**Receiver Delay**	446.9 ns	**Receiver Temp.**	35 °C
Laboratory 2	National Institute of Standards and Technology	**ID Number**	006	**Baseline**	1446.239 km
Latitude	39° 59 min 44.579 s N	**Counter Delay**	0.22 ns	**Reference Source**	UTC(NIST)
Longitude	105° 15 min 43.185 s W	**REF Delay**	435.4 ns	**Mask Angle**	10°
Altitude	1646.72 m	**Receiver Delay**	0.4 ns	**Receiver Temp.**	27 °C

Hours in Common-View	Mean Time Offset (ns)	Range (ns)	Frequency Offset	Confidence (r)
4791	0.03	109.96	+7.83 x 10^{-18}	+0.01

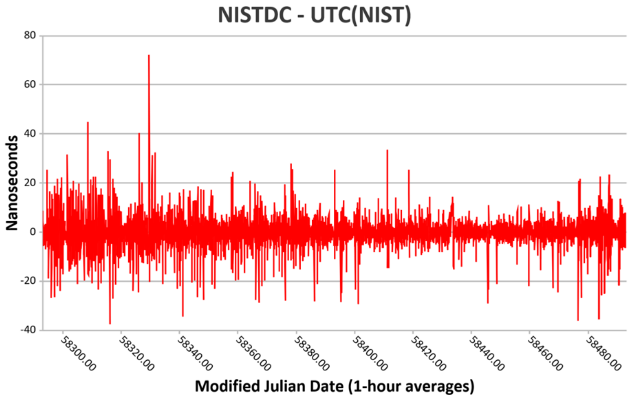


Figure 19. NISTDC v. UTC(NIST) 200-day accuracy chart.

This NISTDC employs the “common-view” technique to keep its replica of UTC(NIST) extremely accurate. Common-view is a simple but effective method for comparing clocks. The time difference between a NISTDC and the UTC(NIST) can be measured by simultaneously comparing both clocks to a signal that is in common view of both sites. The difference between the two comparisons is the time difference between the two clocks. The common-view signal is simply a vehicle used to transfer time from one site to the other. The accuracy of this common-view signal is irrelevant because it does not influence the final measurement result.

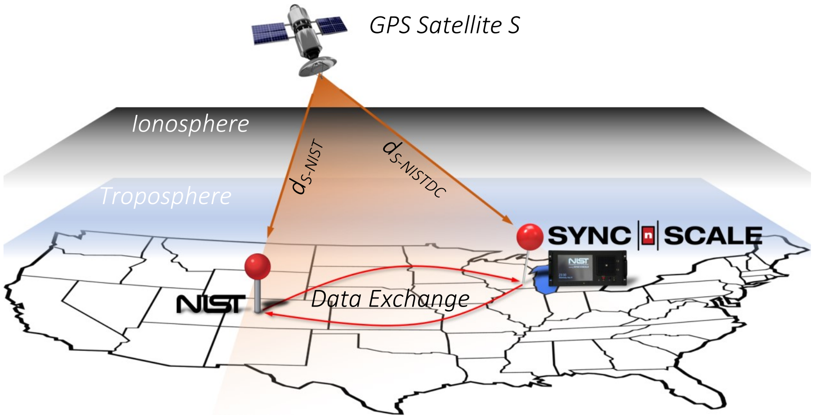


Figure 20. GPS common-view technique.

The GPS satellite S transmits signals that are received at NIST in Colorado and at the Sync-n-Scale NISTDC in Wisconsin (Fig. 20). Both sites compare the received signals to their local clock. The measurement at NIST compares the GPS signals received over the path $d_{S\text{-NIST}}$ to the US time scale, *Clock NIST – S*. The NISTDC compares GPS signals over the path $d_{S\text{-NISTDC}}$ with its rubidium clock, *Clock NISTDC – S*. Delays that are common to both paths $d_{S\text{-NIST}}$ and $d_{S\text{-NISTDC}}$ cancel even if they are not known, but uncorrected delay differences between the two paths add uncertainty to the measurement result. Thus, the basic equation for a common-view GPS measurement is:

$$(\mathit{Clock\ NIST} - S) - (\mathit{Clock\ NISTDC} - S) = (\mathit{Clock\ NIST} - \mathit{Clock\ NISTDC}) + (e_{S\text{-}NIST} - e_{S\text{-}NISTDC})$$

The components that make up the $(e_{S\text{-NIST}} - e_{S\text{-NISTDC}})$ error term include delay differences between the two sites caused by ionospheric and tropospheric delays, multipath signal reflections, environmental conditions, or errors in the GPS antenna coordinates. These factors can be measured or estimated and applied as a correction to the measurement, or they can be accounted for in the uncertainty analysis. It is also necessary to calibrate the GPS receivers used at both sites and account for the delays in the receiver, antenna, and antenna cable. These necessitate the NISTDC being calibrated at NIST prior to field deployment.

UTC(NIST) in Colorado comprises of the primary NIST time scale in Boulder and the secondary NIST time scale in Fort Collins. A new calculated clock difference between the NISTDC rubidium clock and UTC(NIST) in Colorado is taken every 10 minutes. A “steering” correction is sent to the NISTDC rubidium clock. The steering correction is always a dimensionless frequency correction, and time errors are corrected through frequency adjustments **[36]**.

The NISTDC starts steering the rubidium clock automatically as soon as it completes the Windows 10 boot. This active steering of the NISTDC rubidium clock makes it a UTC(NIST) replica.

VI.2. NTP AS A CLOCK OFFSET CALCULATION METHOD IN NISTDC

The NISTDC acts the role of an NTP client. The monitored and measured target system acts the role of an NTP server.

The basic operation of NTP protocol **[37]** is that a client sends a packet to a server and records the time the packet left the client in the Origin Timestamp field (T_1). The server records the arrival time of that packet as the Receive Timestamp (T_2), and assembles the response packet with the Origin Timestamp and Receive Timestamp. The server includes the departure time of the response packet as the Transmit Timestamp when it is sent back toward the client (T_3). The client records the arrival time of the response packet as Destination Timestamp (T_4). At the conclusion of this NTP transaction exchange, the server is in possession of the first three timestamps (T_1, T_2, and T_3), whereas the client is in possession of all four timestamps (T_1, T_2, T_3 and T_4).

Due to the execution complexity of their own software and networking stacks, an NTP transaction between any two equally capable NTP servers in practice is not commutative. When both systems simultaneously initiate an NTP transaction to calculate the clock offset between themselves, the results are contradicting virtually without fail. That is, while the clock offset amount might vary, both would declare identically the other’s time-of-day to be ahead, or behind, compared to its own. That is an impossibility, but which one is incorrect in that particular assertion?

The NISTDC uses the current value of its UTC(NIST) replica as Origin Timestamp (*i.e.* T_1) in the NTP message enquiring the current time of the target system and for subsequent clock offset calculation. The NISTDC does not use its Windows 10 system time-of-day for NTP transactions.

In addition to T_1, T_2, and T_3 which are common artifacts in the NISTDC and its target system, each side also possesses the networking identity (ID) of the other which is used for the NTP packet exchange. A tuple of (ID_{NISTDC}, ID_{TARGET}, T_1, T_2, T_3) encodes the directional semantic of the NTP packet exchange. It is sufficiently unique to allow for mutual corroboration of the clock divergence between the NISTDC and its target system (Fig. 21).

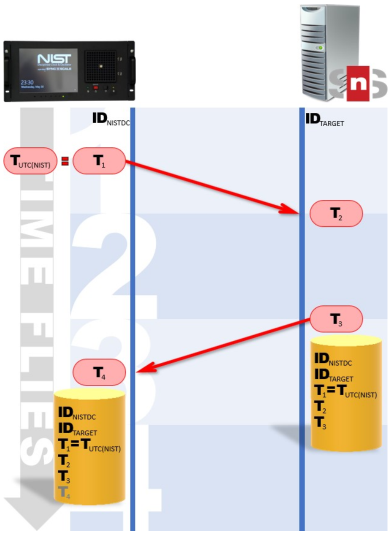


Figure 21. NTP protocol exchanges.

VI.3. NISTDC AS A UTC(NIST) FREQUENCY STANDARD

On the back of the NISTDC chassis (Fig. 22 right), there are three standard 10-MHz frequency outputs, and eight 1 PPS timing outputs. All are synchronized or syntonized to UTC(NIST). These signals can be directly tapped, or distributed farther within the facility.

The GPSDO PCIe interface is designed with its own 1 PPS out MCX port (Fig. 22 left). Measuring the signal synchronicity of this port against that of a NISTDC 1 PPS port (Fig. 7) establishes the frequency stability baseline of the hardware interface and its SXO and OCXO components being used for frequency controls components. Performance of newer parts can be characterized prior to manufacturing production.

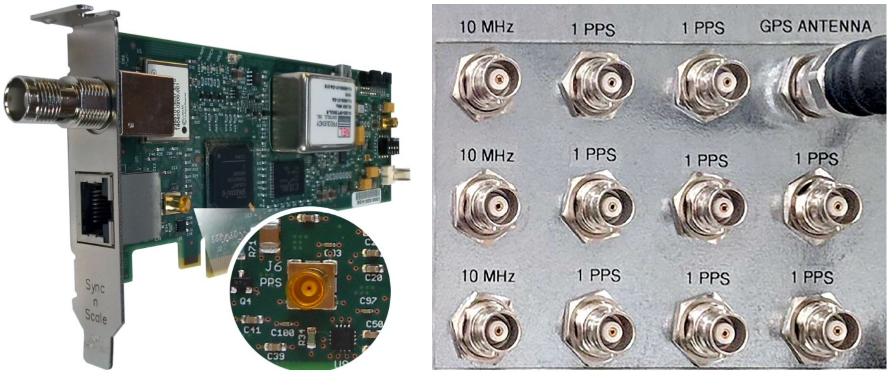


Figure 22. Sync-n-Scale GPSDO and NISTDC timing output ports.

A more complete UTC-accuracy performance characterization of a Windows Server system and its Hyper-V VMs in steady state would emerge from a combination of the above hardware measurements with those from system level software tools discussed earlier. These gained insights would also be valuable during GPS holdover or to understand fault-tolerant server failover performance.

Applying this method, UTC-accuracy of new OS platform capabilities and software improvements can be characterized and quantified prior to production releases. This information allows time-aware workloads and applications higher up in the software stacks to operate with confidence.

VI.4. NISTDC FOR CONTINUOUS UTC-ACCURACY MONITORING AND MEASUREMENT

The NISTDC is fundamentally a time and frequency measurement instrument for IT resources. Not unlike other sophisticate devices that would produce false results when operated without an appropriate harness.

The Sync-n-Scale operated NISTDC has been configured to measure UTC-accuracy of VMs running modern Windows server and desktop OS releases. A harness was constructed to validate the VM IC Time Synchronization performance (Fig. 12). This section discusses the relevant findings from this exercise.

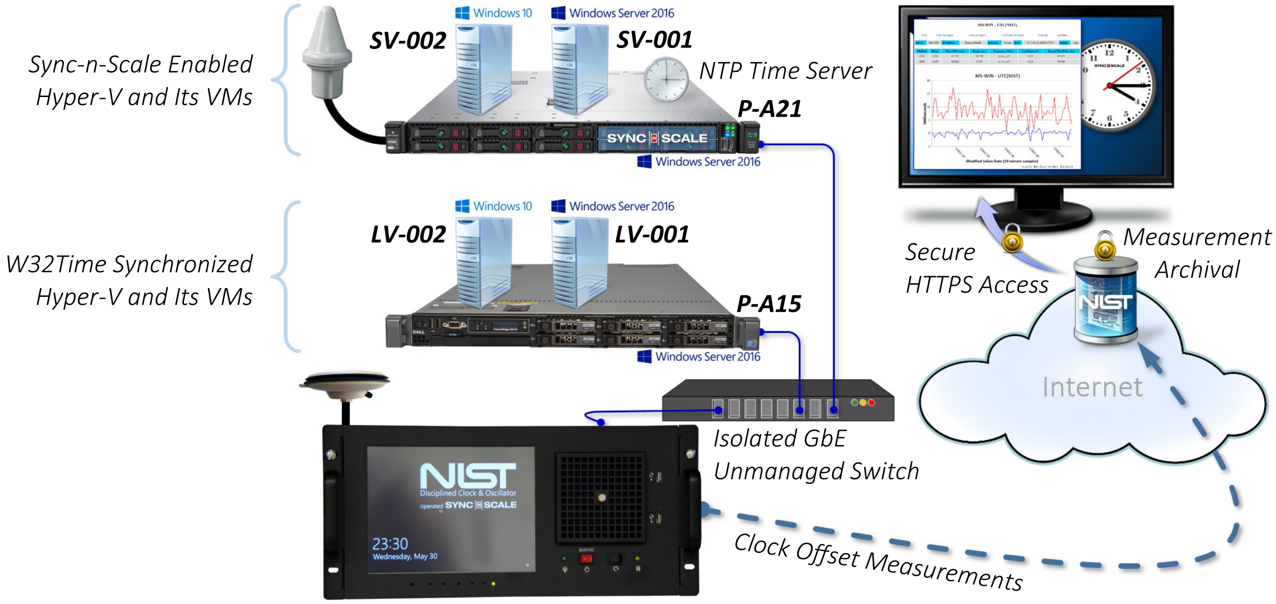


Figure 23. NISTDC measurement harness for physical servers and virtual machines at Sync-n-Scale Burlington, WI site.

In this harness (Fig. 23), all four VMs (**SV-001**, **SV-002**, **LV-001** and **LV-002**) synchronize their clocks with that of their respective Hyper-V platforms (**P-A15** and **P-A21**) using VM IC Time Synchronization. The physical Windows Server **P-A21** is Sync-n-Scale enabled and operates as a W32Time NTP time server. The physical Windows Server **P-A15** synchronizes its clock with the NTP time server **P-A21**. The system OSes are as marked.

In the NTP formula for calculating clock offsets between the NTP time client and its time server, the round-trip delay is assumed to be same. Modern networking gears are sophisticated and will figure out the optimal paths for every data packet based on the current traffic conditions of the interconnected fabric. That cleverness plays against NTP and all other network-based time synchronization techniques. The optimization of flight paths for each packet is totally hidden and out of the control of the two communicating end-points. This harness is focusing on keeping the time it takes for the outgoing and incoming NTP packets to traverse evenly in both directions.

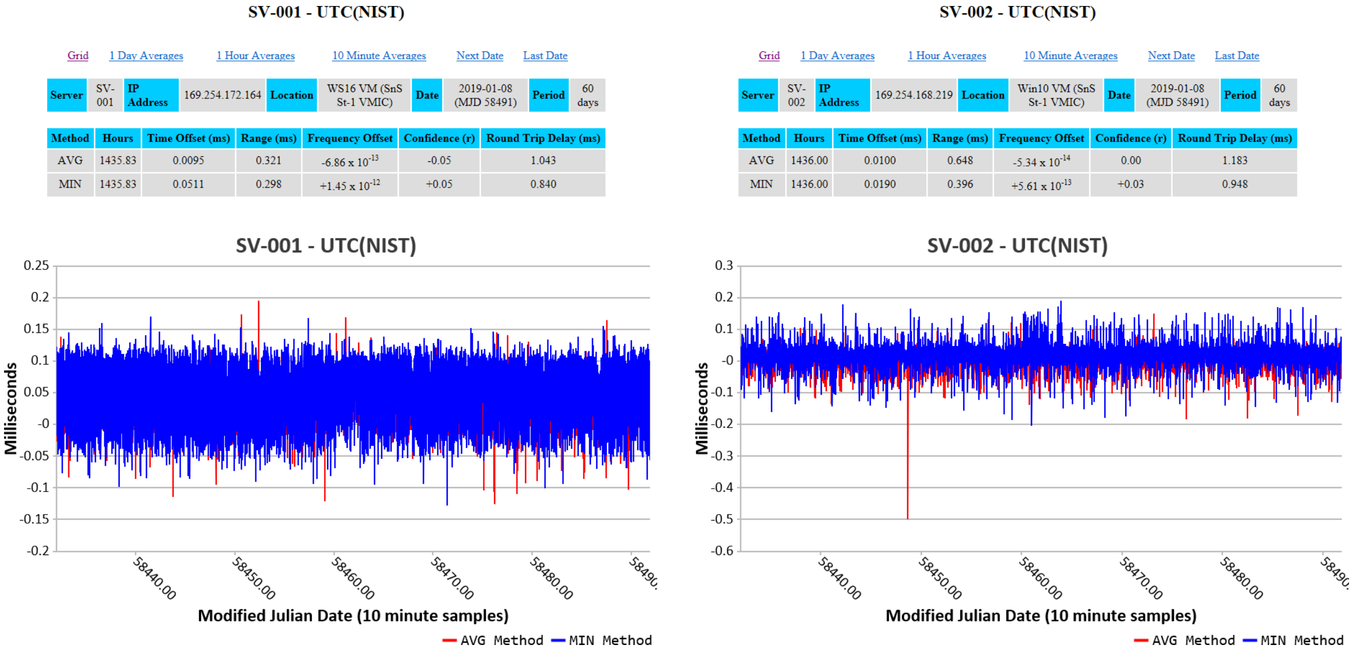
The NISTDC comes with two GbE NIC ports. One is used for internet communications with NIST in Boulder. The second one is used for networking with the on-site monitored and measured systems via an isolated GbE unmanaged switch. This connectivity scheme ensures clock offset measurement NTP packet traffics traverse consistent paths between the NISTDC and its target systems, and avoid nondeterministic asymmetry in their round-trip delays.

The NISTDC acts the role of an authoritative witness and archivist of a system UTC-accuracy. It sends UTC-accuracy measurement data every 10 minutes to a NIST archival repository on the internet. A maximum of 200 days worth of this data can be charted at one time and viewed using any web browser. Each chart shows the UTC-accuracy performance obtained by two measurement and calculation methods: “Average” (AVG) and “Minimum” (MIN).

The AVG method involves comparing a target system to UTC(NIST) replica in the NISTDC every 10 seconds, and then averaging these measurements for 10 minutes (60 measurements). UTC(NIST) replica is available with 100-ns resolution. The difference between the two 10-minute averages is the difference between the time broadcast by the target system and UTC(NIST).

The MIN method compares a target system to UTC(NIST) replica in the NISTDC every 10 seconds for a 10-minute interval (60 measurements). However, only one of the 60 measurements is saved, the one with the shortest round-trip delay. This method saves only 1.67% of the measurements, and is based on the assumption that NTP measurements with the shortest round-trip delays provide the best estimate of the true time difference. UTC(NIST) replica is available with the same 100-ns resolution. The time difference between the target system and UTC(NIST) is obtained by subtracting the two time readings obtained from the measurement with the smallest round-trip delay.

The 60-day (ending on January 8, 2019) UTC-accuracy chart shows the Sync-n-Scale enabled Windows Server 2016 VM **SV-001** (Fig. 24 left) to be nominal at 9.5 µs diverged from UTC(NIST) on average. Another chart of the same time period shows the Sync-n-Scale enabled Windows 10 VM **SV-002** (Fig. 24 right) to be nominal at 10.0 µs diverged from UTC(NIST) on average.



SV-001 - UTC(NIST)

Grid 1 Day Averages 1 Hour Averages 10 Minute Averages Next Date Last Date

Server	SV-001	**IP Address**	169.254.172.164	**Location**	WS16 VM (SnS St-1 VMIC)	**Date**	2019-01-08 (MJD 58491)	**Period**	60 days

Method	Hours	Time Offset (ms)	Range (ms)	Frequency Offset	Confidence (r)	Round Trip Delay (ms)
AVG	1435.83	0.0095	0.321	-6.86×10^{-13}	-0.05	1.043
MIN	1435.83	0.0511	0.298	$+1.45 \times 10^{-12}$	+0.05	0.840

SV-002 - UTC(NIST)

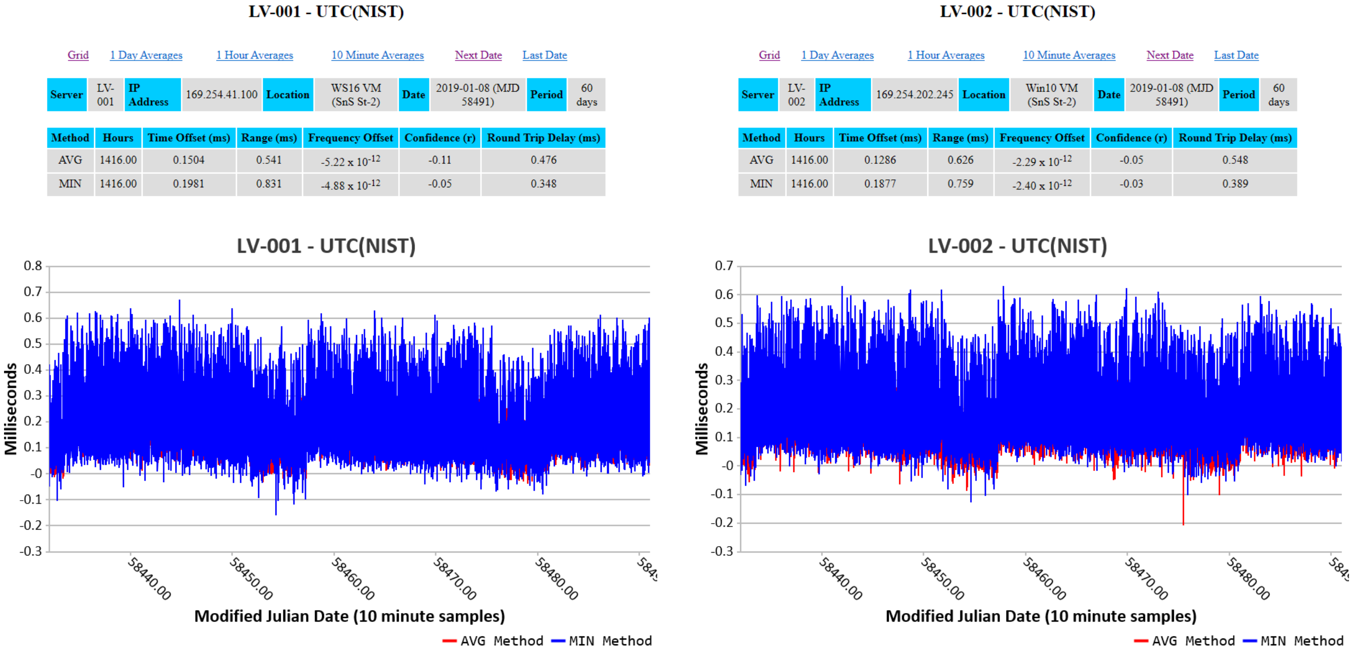
Grid 1 Day Averages 1 Hour Averages 10 Minute Averages Next Date Last Date

Server	SV-002	**IP Address**	169.254.168.219	**Location**	Win10 VM (SnS St-1 VMIC)	**Date**	2019-01-08 (MJD 58491)	**Period**	60 days

Method	Hours	Time Offset (ms)	Range (ms)	Frequency Offset	Confidence (r)	Round Trip Delay (ms)
AVG	1436.00	0.0100	0.648	-5.34×10^{-14}	0.00	1.183
MIN	1436.00	0.0190	0.396	$+5.61 \times 10^{-13}$	+0.03	0.948

Figure 24. NISTDC UTC-accuracy charts of Windows Server 2016 and Windows 10 VMs at Sync-n-Scale Burlington, WI site.

Just as it is important to have a quantified UTC-accuracy precision value, the UTC-accuracy performance graph line pattern over a meaningfully long observation period says a great deal about the underlying clock stability and adjustment mechanisms. A "flat" and "skinny" UTC-accuracy graph line indicating a time difference close to 0 on the X-axis is most desirable. Those graph lines (Fig. 24) illustrate the accuracy and stability of the measured system clock. These are visual cues that would shorten the time to insights for IT professionals.



LV-001 - UTC(NIST)

Grid 1 Day Averages 1 Hour Averages 10 Minute Averages Next Date Last Date

Server	LV-001	**IP Address**	169.254.41.100	**Location**	WS16 VM (SnS St-2)	**Date**	2019-01-08 (MJD 58491)	**Period**	60 days

Method	Hours	Time Offset (ms)	Range (ms)	Frequency Offset	Confidence (r)	Round Trip Delay (ms)
AVG	1416.00	0.1504	0.541	-5.22×10^{-12}	-0.11	0.476
MIN	1416.00	0.1981	0.831	-4.88×10^{-12}	-0.05	0.348

LV-002 - UTC(NIST)

Grid 1 Day Averages 1 Hour Averages 10 Minute Averages Next Date Last Date

Server	LV-002	**IP Address**	169.254.202.245	**Location**	Win10 VM (SnS St-2)	**Date**	2019-01-08 (MJD 58491)	**Period**	60 days

Method	Hours	Time Offset (ms)	Range (ms)	Frequency Offset	Confidence (r)	Round Trip Delay (ms)
AVG	1416.00	0.1286	0.626	-2.29×10^{-12}	-0.05	0.548
MIN	1416.00	0.1877	0.759	-2.40×10^{-12}	-0.03	0.389

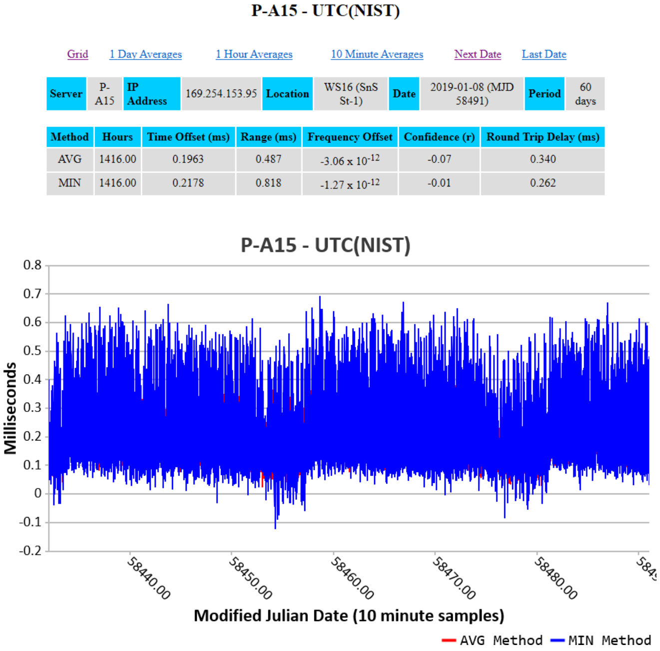
Figure 25. NISTDC UTC-accuracy charts of Windows Server 2016 and Windows 10 VMs at Sync-n-Scale Burlington, WI site.

The next set of 60-day (also ending on January 8, 2019) UTC-accuracy charts quantify the precision of Windows Server 2016 VM **LV-001** (Fig. 25 left) to be nominal at 150.4 µs diverged from UTC(NIST) on average; and that of Windows 10 VM **LV-002** (Fig. 25 right) to be nominal at 128.6 µs diverged from UTC(NIST) on average.

The quantified differences in these charts (Figs. 24 and 25) are also visually pronounced because of the 60-day worth of measurements being examined. A shorter measurement or charting period might not have yielded similar insights.

Despite the fact that the VMs **LV-001** and **LV-002** (Fig. 25) are operating identical Windows OS bits (copies of the same Hyper-V VHD OS images), their UTC-accuracy on average were 10+ times worse than that of the other VMs **SV-001** and **SV-002** (Fig. 24). This performance difference between the two sets of VMs (Fig. 23) was during the same time period, in the same harness and under the same operating conditions, except for the mechanism used by their respective Hyper-V for disciplining its own clock.

A closer look at the Hyper-V host **P-A15** UTC-accuracy is warranted (Fig. 26) because this Windows Server 2016 system relies on an NTP time server (**P-A21**) for its clock accuracy. It is operating in a favorably constructed harness according to guidance in the blog article Accurate Time for Windows Server 2016 **[38]**. The outlined critical factors for accurate time were adequately met, *i.e.* solid source clock, stable client clock, and symmetrical NTP communication.



P-A15 - UTC(NIST)

Grid 1 Day Averages 1 Hour Averages 10 Minute Averages Next Date Last Date

Server	P-A15	**IP Address**	169.254.153.95	**Location**	WS16 (SnS St-1)	**Date**	2019-01-08 (MJD 58491)	**Period**	60 days

Method	Hours	Time Offset (ms)	Range (ms)	Frequency Offset	Confidence (r)	Round Trip Delay (ms)
AVG	1416.00	0.1963	0.487	-3.06×10^{-12}	-0.07	0.340
MIN	1416.00	0.2178	0.818	-1.27×10^{-12}	-0.01	0.262

Figure 26. NISTDC UTC-accuracy chart of Windows Server 2016 physical server at Sync-n-Scale Burlington, WI site.

The 60-day UTC-accuracy of Hyper-V host **P-A15** was nominally at 196.3 µs diverged from UTC(NIST) on average. Because of this time accuracy performance, all NTP clients of **P-A15** at lower strata cannot hope to perform better. Nevertheless, the UTC-accuracy performance of **P-A15** and that of its VMs **LV-001** and **LV-002** during this time period was still wildly much better than had they been configured to keep the default **time.windows.com** as the NTP time source.

VI.5. NISTDC AT DATA CENTER AND PLANET SCALES

At this point, we have described the NISTDC as an instrument for monitoring and measuring UTC-accuracy of Windows physical systems and their virtual machines at server rack scale. By design, its capabilities are identical at data center and planet scales owing to the general uniformity of GPS as a global reference time source.

This property makes the NISTDC also an independent authoritative instrument for continuous UTC-accuracy monitoring of planet-scale distributed workloads and solutions across data centers to ensure their optimal performance. Sync-n-Scale enabled general-purpose computing platforms have been shown capable of operating these workloads at the highest level of UTC-accuracy precision. Being persistently UTC-accurate means natural synchronicity without developers, implementors and support IT professionals being heroic.

Blockchain consortium is an example needing UTC-accuracy for technical compliance, and monitoring for operational awareness. Distributed database is another example. These needs are same at rack, data center and planet scales.

VI.6. HIGH-PERFORMANCE PLANET-SCALE DATABASES

Time-series data are generally timestamped measurements or events that are tracked, monitored, aggregated and stored over time. Time-series data applications are proliferating mostly due to digital transformation of industries to take advantage of machine learning and other advances in data science.

A distributed database stores copies of data in multiple locations in order to efficiently receive and deliver them. These database location instances are linked to allow access and updates to these copies of data as one.

Modern IT solutions typically are geographically global in nature. Their applications and databases are geographically distributed for high availability and for maximum throughput. The computing platforms maintain clock synchronicity to preserve the integrity of time-series data being gathered by the applications. The linked data stores maintain clock synchronicity across global distances to efficiently operate as high-performance databases. All need high-speed network connectivity among these locations.

Sync-n-Scale enabled persistent UTC-accuracy across the compute and storage platforms ensures their natural synchronicity at planet scale. NISTDC corroborated UTC-accuracy measurements provide continuous operational awareness and insights for performance diagnosis.

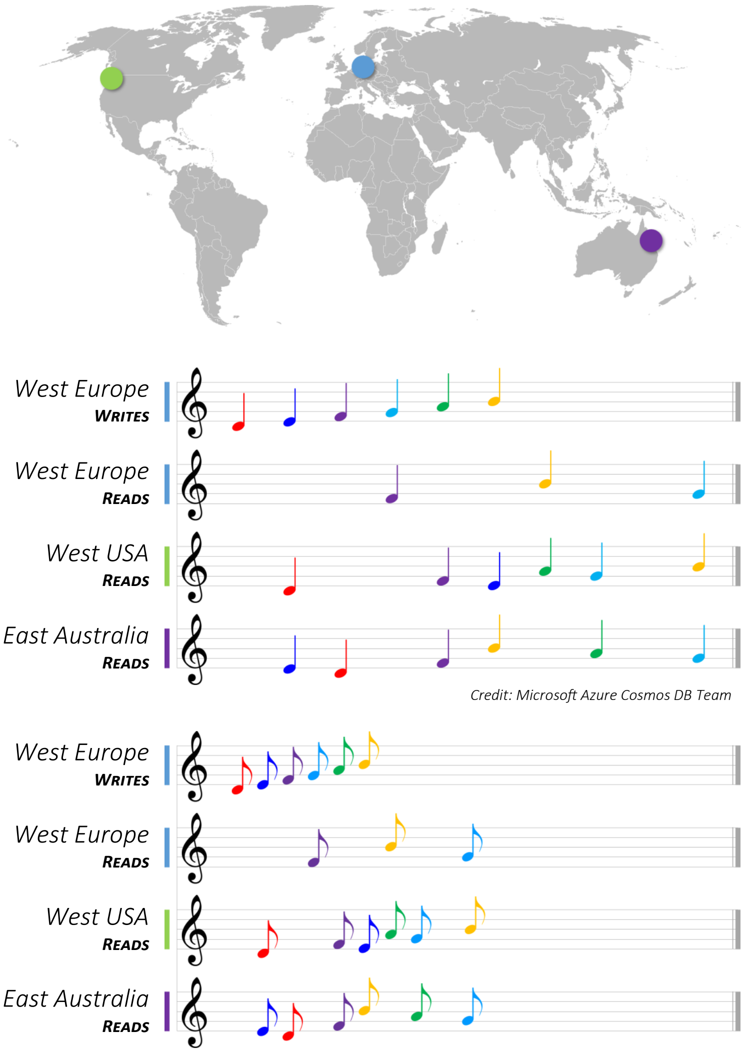
The orchestral music score metaphor **[39]** is one way to illustrate this high-performance potential gain. The example IT solution is operated out of three data centers (Fig. 27 top) in the US, EU and Australia. Its time-series distributed database spans across these three locations.

Data written to its West Europe data store can be read by applications at other two locales in the order represented by the music notes on the staffs denoting the different sites (Fig. 27 middle). The quarter note tempo represents one level of synchronicity precision across the three sites.

Transaction order can be established quicker when there is tighter synchronicity (UTC-accuracy) precision across the platforms. These musical notes could now be "played" at a faster eighth note tempo (Fig. 27 bottom). This represents the data being available for reads outside of its write location faster across the distributed database. This performance gain benefits modern IT solutions in all industry sectors, not just financial services.

VII. LOCATION, LOCATION, LOCATION

Location awareness is no longer a feature reserved for mobile devices and their applications. Mobile IT platforms and their deployments are no longer rarities, or niche solutions preferred by selected industries such as transportation and defense. These trends have been accelerated to address distributed IT and the Internet of Things across industries.



Credit: Microsoft Azure Cosmos DB Team

Figure 27. Orchestral Music Score Metaphor.

Timekeeping does not have to be a stand-alone capability. A combination of persistent UTC-accuracy and awareness of true geolocation would enable a wide range of new platform capabilities for modern IT solutions. Sync-n-Scale hardware interfaces are capable of exploiting the full complement of GPS signals for positioning and timing, and delivering them to the OS and its hypervisor for general-purpose computing applications. Current Geolocation API **[40]** support in Windows OS is uneven between the in-market server and desktop editions. Unlike the VM IC Time Synchronization support, there is not yet a similar mechanism for Hyper-V VMs or comparable platform abstractions such as Containers to obtain their true geographical location.

When properly aggregated, the yielded functionalities are greater than the sum of these discrete capabilities. Cloud-based applications can be demonstrated to operate in compliance with data sovereignty regulatory requirements. Cost effective software defined mitigation for GPS-denied conditions can be devised to operate without exotic hardware.

VII.1. DATA SOVEREIGNTY COMPLIANCE

A data center is where application software and customer data running on the software are located. Vendors of cloud-based IT services must maintain transparency of where they replicate customer data at any given time for protection against failure or local disaster. If a data center ceases functioning for any reason, customer data will not be lost if the application software and customer data running on that application software are also available from a second or possibly third data center. And assuming it works smoothly enough, customers might not even be notified when such a failover occurs. Depending on the particular service, failover may not result in any service interruption at all.

Global enterprises have leveraged the internet and cloud-based computing services, along with data centers, to establish private communication networks and capture efficiencies from global technology. As a result of such globalization, in recent years, many nations have issued geographic location rules restricting how corporations can handle and transmit their customers' data across borders, including through these private networks. Certain entities require that specific types of data, for example government data, employee data, or telecommunications traffic data be stored within a limited geographical border, and in some cases, such data may even not be accessed from outside of a geographical border.

True GPS-originated information can be used to determine if specific applications, data stores, virtual machines or comparable platform abstractions are definitively inside particular geofences before being allowed to operate. Continuous records of both positioning and timing during their operations would allow developers and IT professionals to gain insights into the reliability, availability, serviceability and locality of their cloud-based IT components.

VII.2. SOFTWARE DEFINED MITIGATION FOR GPS-DENIED CONDITIONS

GPS positioning and timing data services are critical to industry sectors such as energy (*e.g.* power grids), financial services (*e.g.* banking and market transaction systems) and information technology (*e.g.* data centers). As these critical infrastructure assets, networks, and systems have become much more interdependent across vast regions, crossing jurisdictional and national boundaries, and time zones, the need for operational awareness of accurate and precise timing services is vital to their continued functioning.

Worldwide outage of GPS infrastructure caused by man-made calamity and conflicts is outside the scope of this paper. Local and regional GPS disruption is more common. This section proposes a possible civil defense mitigation approach to be built on UTC-accuracy and geolocation awareness of IT resources in the private sectors.

In order to detect GPS-denied conditions or timing manipulation, we must be able to ascertain what is nominally UTC-accurate time at all time, and from a trusted reference time source. We are proposing an ensemble of UTC-accurate and geolocation-aware (*e.g.* Sync-n-Scale enabled) systems operated in tandem with participating NISTDC units, at NIST and privately operated elsewhere in the world (Fig. 28). These NISTDC units might be located at commercial data centers, or public premises such as college campuses, or local government facilities.

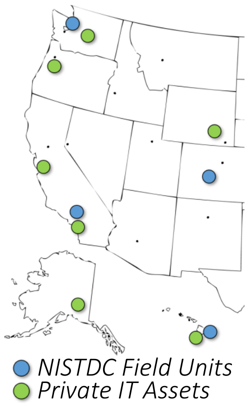


Figure 28. Networked sensors for GPS disruption warning.

Collectively these systems form mesoscale networks of "sensors" for detecting GPS disruption and timing manipulation. As a discreet and fungible network of sensors, they would transmit their own operating positioning and timing information, and the full complement of GPS ephemeris data as received by each. These data points are then aggregated in cloud space for machine learning and deep learning analytics.

Aggregated sensor data and their analytical logics are protected and access controlled in cloud space. Reported GPS ephemeris data or their absence from these sensors can be analyzed in real time to identify the specific sensors being affected by GPS disruption or subjected to timing manipulation, abruptly or gradually over time. Size of the affected locale or region can be determined from the aggregated geolocations of affected sensors.

VIII. SUMMARY

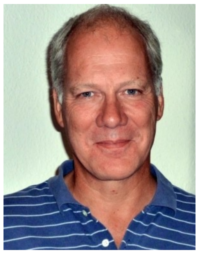
Timekeeping for modern IT solutions can be more effective by bringing an authoritatively curated time source directly into general-purpose computing platforms to make them persistently UTC-accurate; refining tools, methods and harnesses to help IT professionals demonstrate that their IT solutions are UTC-accurate; and exploiting these new capabilities fully in future solutions.

IX. ABOUT THE AUTHORS

Son VoBa joined Sync-n-Scale (and its parent company NEL Frequency Controls, Inc.) in 2014 as the Director of Cloud & Enterprise Solutions following a 15-year stint in technical program management positions at Microsoft Corporation in Windows Server and related organizations. Prior to joining Microsoft in 2000, Mr. VoBa had spent 20 years at Digital Equipment Corporation (DEC), which later was acquired by Compaq Corporation. He graduated from Purdue University in West Lafayette, Indiana with degrees in Science and Computer Science. His work at Sync-n-Scale and Microsoft resulted in granted and pending patents in the areas of basic computing, applied RFID in data center infrastructure, virtual appliance distribution, and applied cryptography for trustworthy computing.

Charles Ulland is the Managing Director of Sync-n-Scale. In addition to this role, Mr. Ulland also serves as a Managing Director at NEL Frequency Controls, Inc., a company he has been affiliated with since 1990. He did his undergraduate study at Minnesota State University at Mankato, and completed his Master of Business Administration (MBA) at University of Wisconsin in Milwaukee.

Michael Lombardi is a metrologist in the Time and Frequency Division of the National Institute of Standards and Technology (NIST) in Boulder, Colorado. He is the project manager for the NIST Frequency Measurement and Analysis (FMAS) and Time Measurement and Analysis (TMAS) remote calibration services and the Time and Frequency Division Quality Manager. Mr. Lombardi served as the chair of the Time and Frequency Metrology Working Group of the Interamerican Metrology System (SIM) from 2011 to 2015 and as the program chair of PTTI in 2017. He has published over 100 papers related to time and frequency metrology.



Arno Lentfer embarked on his one-man multi-year Windows Timestamp Project a decade ago. He is the owner and curator of its independent repository of deep-dive knowledge about how Windows OS keeps time, and the G Suite tools which had been around a great deal longer than the recently renamed Google Apps (to also be G Suite). Mr. Lentfer received patents in the fields of X-ray lithography and goniometric systems. He co-founded X-ray Research GmbH in 1990 and spent the next 15 years developing these technologies before retiring to become a freelance hardware and software designer.

X. ACKNOWLEDGMENTS

This paper includes contributions from the U. S. government, and as such, is not subject to copyright. The use and mention of commercial products does not imply endorsement by NIST.

XI. REFERENCES & ADDITIONAL RESOURCES

[1] FINRA Disciplinary Actions against Dealerweb Inc., Broker-Dealer CRD No. 19662, Case ID 20160507843-01 (page 2): *"Without admitting or denying the findings, the firm consented to the sanctions and to the entry of findings that it failed to report the correct trade execution time for transactions in Trade Reporting and Compliance Engine® (TRACE®)-eligible agency debt securities to TRACE, failed to report the same transactions to TRACE in TRACE-eligible agency debt securities within 15 minutes of the execution time, and failed to show the correct execution time on brokerage order memoranda."*

[2] FINRA Disciplinary Actions against Deutsche Bank Securities Inc., Broker-Dealer CRD No. 2525, Case ID 20130379938-01 (page 3): *"The findings also stated that the firm's time stamps reported that ISOs had been routed two to six seconds after the trades were executed. These time stamping inaccuracies occurred because they were not being properly recorded for ISO sweep orders."*; Case ID 20140418941-01 (page 9): *"In addition, the firm failed to report the correct time of trade to the RTRS for these reports of transactions in municipal securities. The firm failed to show the execution time on brokerage orders' memoranda. The firm failed to include the "seconds" in the time of trade field in its transaction reports to the RTRS in reports."*, and Case ID 20150443249-01 (page 6): *"The findings stated that the firm did not enforce certain of its WSPs because it failed to adequately escalate TRACE reporting deficiencies, including, but not limited to, reporting the incorrect time of execution, late reporting, dealer-mismatch reporting issues, and setting up new issues."*

[3] The DHS identified critical sectors (https://www.dhs.gov/cisa/critical-infrastructure-sectors) include Chemical, Commercial Facilities, Communications, Critical Manufacturing, Dams, Defense Industrial Base, Emergency Services, Energy, Financial Services, Food and Agriculture, Government Facilities, Healthcare and Public Health, Information Technology, and Nuclear Reactors, Materials, and Waste.

[4] A. Malhotra, I.E. Cohen, E. Brakke, and S. Goldberg, "Attacking the Network Time Protocol," Boston University, Department of Computer Science (2015).

[5] M.A. Weiss, J. Eidson, C. Barry, D. Broman, L. Goldin, R. Iannucci, E.A. Lee, and K. Stanton, "Time-Aware Applications, Computers, and Communication Systems (TAACCS)," *NIST Technical Note 1867* (2005).

[6] AJ. Dellinger, "Microsoft's Time.Windows.com Causes Computers To Display Wrong Time," *International Business Times* (April 3, 2017).

[7] Google Public NTP (https://developers.google.com/time/) web site states, *"Since 2008, instead of applying leap seconds to our servers using clock steps, we have "smeared" the extra second across the hours before and after each leap. The leap smear applies to all Google services, including all our APIs."* AWS Adjusted Time is described at its News Blog (https://aws.amazon.com/blogs/aws/look-before-you-leap-the-coming-leap-second-and-aws/), *"Some organizations, including Amazon Web Services, plan to spread the extra second over many hours surrounding the leap second by making every second slightly longer."*

[8] J. Levine, "UTC and NIST Time," *ATIS Presentation* https://youtu.be/DNdWq-ADSoM?t=1547 (2017).

[9] A Microsoft Windows cluster-aware application calls the cluster APIs to determine the context under which it is running and can failover between nodes for high availability.

[10] Commercial RF over Fiber solutions are available in the marketplace for installations requiring longer distance between GPS antenna and Sync-n-Scale GPSDO PCIe hardware interface.

[11] https://docs.microsoft.com/en-us/windows-server/networking/windows-time-service/windows-time-service-top

[12] https://docs.microsoft.com/en-us/windows/desktop/SysInfo/time-provider

[13] https://docs.microsoft.com/en-us/windows-hardware/test/wpt/windows-performance-analyzer https://www.microsoft.com/en-us/p/windows-performance-analyzer/9n0w1b2bxgnz?activetab=pivot:overviewtab

[14] https://www.youtube.com/watch?v=zxFd_QZbT8s

[15] https://www.hpe.com/us/en/solutions/cloud/azure-hybrid-cloud.html

[16] D. Matsakis, J. Levine, and M.A. Lombardi, "Metrological and Legal Traceability of Time Signals," *Proceedings of 2018 ION Precise Time and Time Interval Meeting (PTTI)*, at Reston, Virginia.

[17] https://www.hpe.com/us/en/solutions/transform-hybrid.html

[18] https://aws.amazon.com/outposts/

[19] https://docs.microsoft.com/en-us/virtualization/hyper-v-on-windows/reference/hyper-v-architecture

[20] https://docs.microsoft.com/en-us/virtualization/hyper-v-on-windows/reference/integration-services

[21] https://docs.microsoft.com/en-us/windows-server/virtualization/hyper-v/Supported-Linux-and-FreeBSD-virtual-machines-for-Hyper-V-on-Windows

[22] https://www.microsoft.com/en-us/download/details.aspx?id=55106

[23] https://docs.microsoft.com/en-us/virtualization/hyper-v-on-windows/reference/integration-services#hyper-v-time-synchronization-service

[24] https://docs.microsoft.com/en-us/azure/virtual-machines/linux/time-sync#check-for-ptp

[25] https://docs.microsoft.com/en-us/virtualization/windowscontainers/deploy-containers/deploy-containers-on-server#install-base-container-images

[26] https://github.com/Microsoft/Windows-Time-Calibration-Tools

[27] https://github.com/Microsoft/Windows-Time-Calibration-Tools/tree/master/OsTimeSampler

[28] The Windows Timestamp Project (http://www.windowstimestamp.com/) is not affiliated with, nor has it been authorized, sponsored, or otherwise approved by Microsoft Corporation.

[29] https://docs.microsoft.com/en-us/azure/virtual-machines/windows/time-sync

[30] https://www.finra.org/sites/default/files/notice_doc_file_ref/Regulatory-Notice-16-23.pdf

[31] https://www.bipm.org/en/worldwide-metrology/national/

[32] CRADA CN-15-0081.

[33] M. Weiss, G. Petit, and Z. Jiang, 2005 "A Comparison of GPS Common-View Time Transfer to All-in-View," *Proceedings of the 2005 IEEE International Frequency Control Symposium and Exposition*, 5 pp.. 10.1109/FREQ.2005.1573953.

[34] https://docs.microsoft.com/en-us/windows/security/information-protection/bitlocker/bitlocker-overview

[35] https://www.nist.gov/pml/weights-and-measures/si-units-time

[36] M. Lombardi, A. Novick, B. Cooke, and G. Neville-Neil, 2016,"Accurate, Traceable, and Verifiable Time Synchronization for World Financial Markets," *Journal of Research of the National Institute of Standards and Technology*, 121, 436-463.

[37] D. Mills, 2011, “Computer Network Time Synchronization: the Network Time Protocol on Earth and in Space, Second Edition,” CRC Press, 466 pp.

[38] https://docs.microsoft.com/en-us/windows-server/networking/windows-time-service/accurate-time

[39] https://languye-webapp.azurewebsites.net/music/music.html

[40] https://docs.microsoft.com/en-us/windows/uwp/maps-and-location/get-location