NANOSECOND RANGE WINDOWS ACCURACY WITH TIMEKEEPER

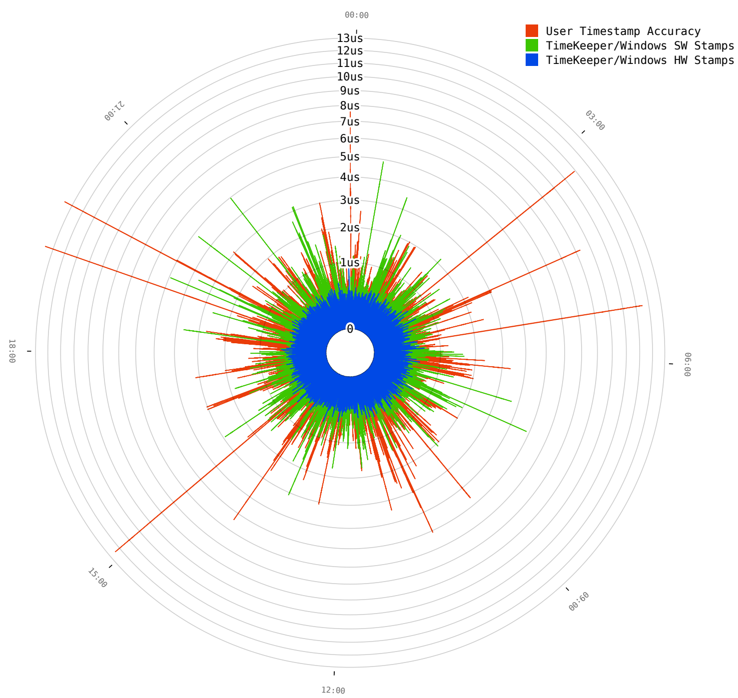
FSMLABS RESULTS OF TESTING EXPERIMENTAL WINDOWS TIMESTAMPING FEATURES

Matt Sherer
FSMLabs Inc. sherer@fsmlabs.com

July 7, 2020

ABSTRACT

FSMLabs using Microsoft features has enabled nanoseconds-range UTC accuracy on Windows for both PTP and NTP without the need for dedicated hardware. This paper outlines preliminary testing results performed by FSMLabs. [1]



Zeroing in - 24 Hours of TimeKeeper/Windows Accuracy with Hardware Timestamping

1 TimeKeeper and Windows

TimeKeeper has long supported Windows, and has improved along with the platform. Its PTP and NTP accuracy already far exceed what's needed to satisfy regulatory compliance requirements, allowing users to make intelligent

[1]When released by Microsoft, TimeKeeper will leverage these capabilities to enable nanosecond range accuracy and improved software timestamping on Windows. Microsoft did not participate in or independently validate the testing described in this paper.

decisions based on having the correct time. With the work in this paper, TimeKeeper pushes Windows accuracy further, adding timestamping features that were previously only supported under Linux.

Good accuracy was already possible on Windows, and now even more so. Using these new improvements, TimeKeeper can even more clearly disprove the following common misconceptions in time sync:

- NTP (especially on Windows) can only be accurate to milliseconds [1]
- You need to choose either PTP or NTP for your client timing [3]
- Windows systems generally don't have accurate time [2]

Before we look at these though let's go over what's new here and why it's important.

Having accurate time on a client is only possible if you have accurate time at your time source and you know how long it takes to get that time to the client. The client then factors out that delay so it can be in sync with the time source. Both NTP and PTP have mechanisms to handle this very well. The farther a packet is away from the wire when a timestamp is taken, the noisier that delay calculation will be. Much of that noise can be handled with advanced filtering like what TimeKeeper does.

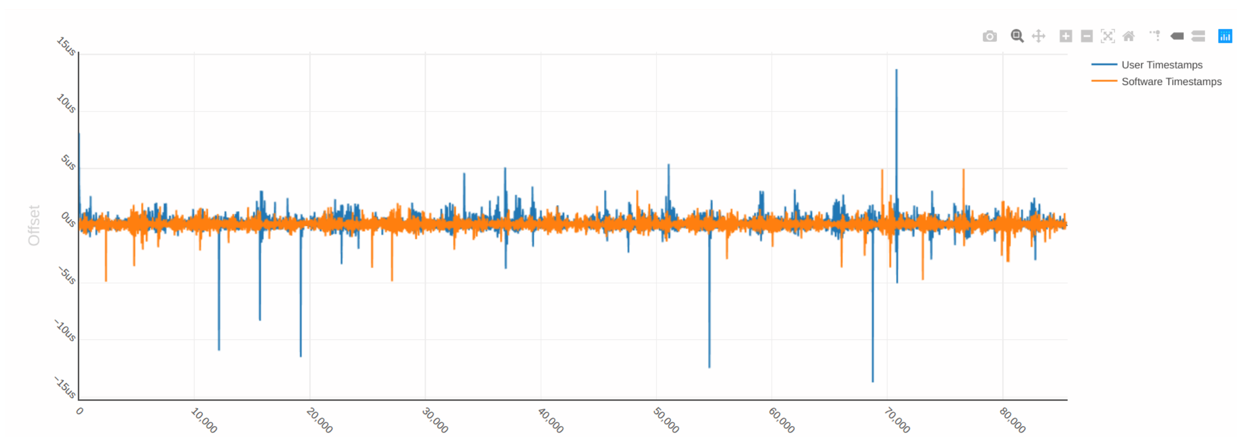
Getting those timestamps taken 'closer to the wire' by the operating system removes delay noise on the host, whether it's done by the operating system as the packets traverse the stack (software timestamps) or by hardware as packets leave or enter the NIC. Linux support for this is well established and helps TimeKeeper get accuracies in the low nanosecond range.

Now these capabilities are coming to Windows and with it, low nanosecond accurate sync.

2 Nanosecond Accuracy with NTP

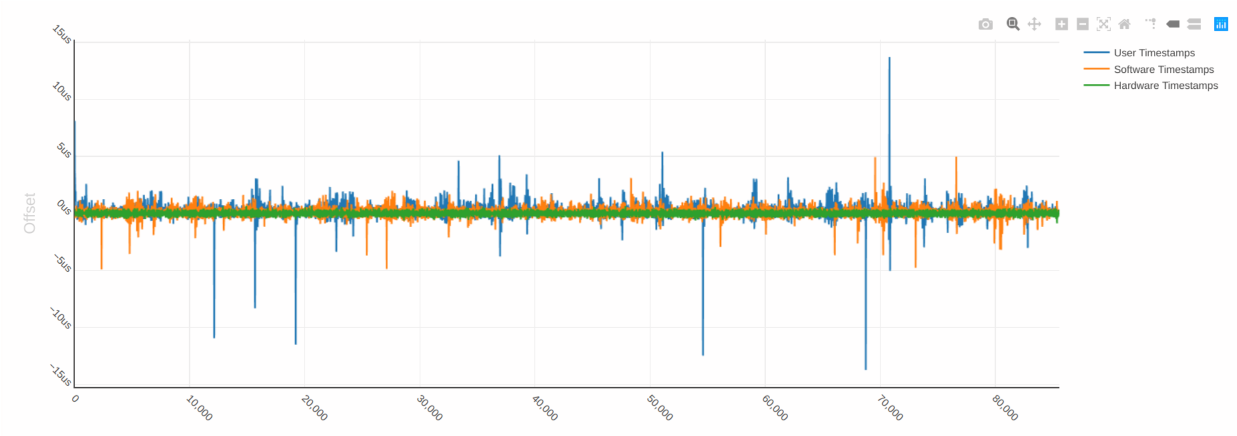
Let's look at improvements in accuracy with NTP first. That's what is shown above, with user-collected timestamp accuracies in red. Note that they're subject to the worst spikes due to scheduling and other delays. That same plot also shows software timestamps in green. It's clear that the software timestamps are less susceptible to noise.

Said another way, you can clearly see that the software timestamps still include noise, but are better bounded. Below we see clock accuracy calculated with and without software timestamps from Windows:



TimeKeeper's doing very well with its own cross check timestamps (in blue), but you can see it does even better if it also has software timestamps (in orange) to work with too. Now TimeKeeper customers can get a better bound on their timestamp accuracy on Windows no matter what hardware they've deployed.

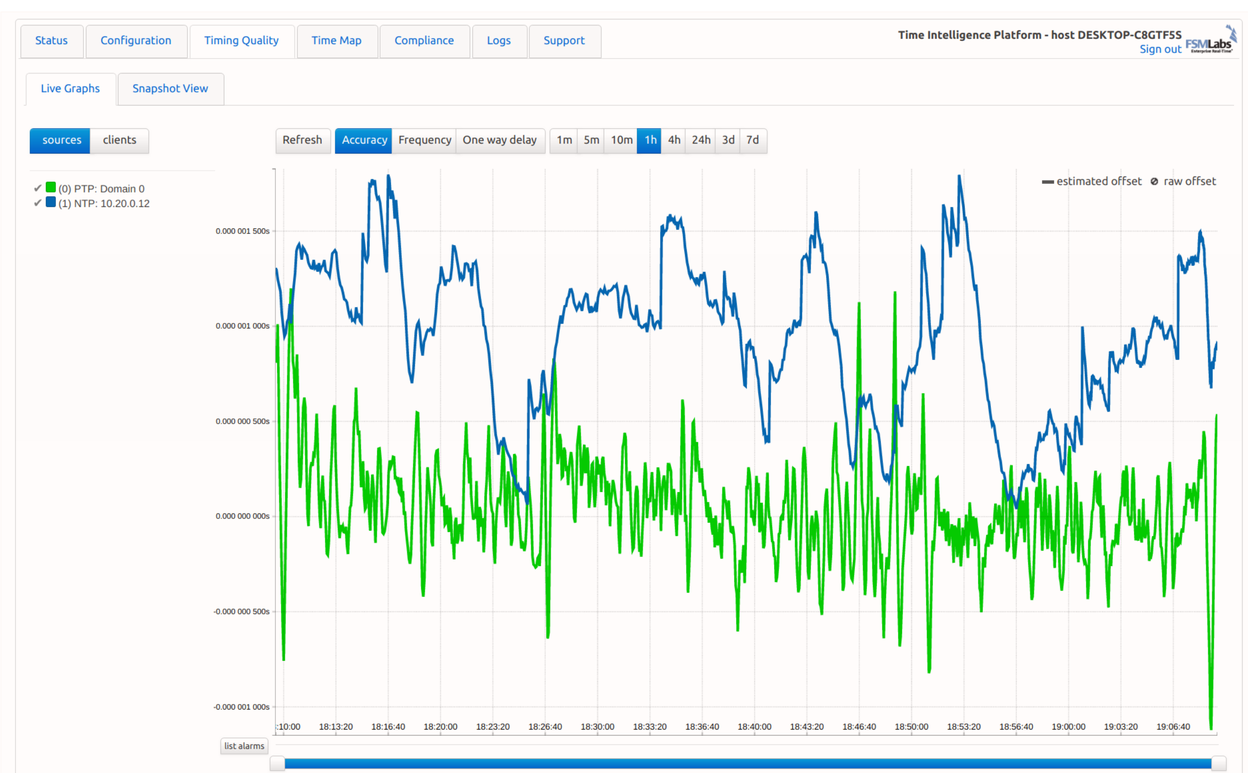
With hardware timestamps TimeKeeper can take accuracy even further, to the sub microsecond/low nanosecond range which was previously only an option on Linux. TimeKeeper with hardware timestamps can maintain a very accurate system clock as seen below, showing user timestamps, software timestamps, and hardware timestamps too (in green). Note how stable the clock is with hardware timestamps - over the entire test it's inside the 1 microsecond mark.



3 Equivalent Accuracy with NTP or PTP

That sub-microsecond accuracy above is not limited to NTP. There's nothing specific to it as a protocol here, PTP can do just as well. Where you can get 500 nanosecond accurate Windows time via NTP, you can do the same with PTP[2].

Here you can see both NTP and PTP being used to get a good sync on a client, with a higher rate PTP on domain 0 as the primary source and NTP as the backup. This is plotted with the standard TimeKeeper web GUI, showing exactly what customers can see in production. In this case TimeKeeper's able to identify a small discrepancy between the PTP and NTP data so it can be investigated.



[2] Subject to the accuracy of your NTP server or PTP grandmaster.

4 Windows Accuracy Competitive with Linux

TimeKeeper customers are used to being able to get accuracy on Windows in the low microsecond range, but where higher accuracy was needed, they used Linux. With these improvements, you can get better accuracy on Windows with just software timestamps, improving baseline accuracy, latency calculations, and so on. If needed, hardware timestamping can push things further, into areas that used to be just the domain of Linux.

Once released by Microsoft, all that's needed for this will be:

- TimeKeeper with support for these features built in. [3]
- Windows Server OS/Windows Client OS with support for hardware and/or software timestamping.
- If hardware timestamps are needed, a NIC that supports hardware timestamping on Windows.

There's no need for custom/dedicated cards or cabling, GPS receivers on your Windows hosts, or really anything beyond what you may already have deployed. With just TimeKeeper and optionally an off the shelf NIC, you've got a Windows client that's accurate to less than a microsecond to UTC for reporting, accurate trading decisions, latency validation, or anything else that you may have needed Linux for before.

4.1 Serving Accurate PTP and NTP

Beyond just acting as an accurate client, TimeKeeper can also serve NTP and PTP just like on Linux, acting as a boundary clock, stratum server, or client data aggregator for Compliance reporting. With an accurate clock, delivering accurate PTP to downstream hosts or remote NTP clients just takes a few clicks.

Below we can see just how accurate Windows really is by comparing its time with a GPS reference signal. This is from the perspective of a TimeKeeper Grandmaster taking in GPS, serving that to Windows via NTP and PTP, and then getting the served PTP and NTP from Windows back for comparison. The entire scale is 1 full microsecond:

This is not a short burst of accurate time either - it's a full 24 hour period of testing where time went from GPS via NTP/PTP to Windows and back all without deviating by more than a single microsecond.

[3] Contact sales@fsmlabs.com for details.

If you'd like to see how this would fit in your environment, contact us at sales@fsmlabs.com.

References

[1] FSMLabs. "120 Nanosecond Worst Case NTP Performance with TimeKeeper". In: (2018 (accessed April 2020)). "". URL: https://fsmtime.com/blog/article/ntp_performance/.

[2] Microsoft. "Windows Time service tools and settings". In: (2020 (accessed April 2020)). "". URL: https://docs.microsoft.com/en-us/windows-server/networking/windows-time-service/windows-time-service-tools-and-settings/.

[3] *The business case for being time protocol agnostic*. https://fsmtime.com/blog/article/the-business-case-for-being-time-protocol-agnostic-/. (Accessed on 04/16/2020).