Cloud clocksync showdown: Ntpd vs Chrony vs Clockwork

July 13, 2022 · clockwork.io/blog/cloud-clocksync-showdown-ntpd-vs-chrony-vs-clockwork/

TL;DR:

- Distributed systems in the cloud increasingly rely on accurately synchronized clocks for correctness guarantees and peak performance.
- We evaluate the accuracy and reliability of three clock sync systems: Standard ntpd vs. Chrony with cloud-optimized configuration vs. Clockwork. The NTP-based solutions are client-server based, while Clockwork's algorithm uses a probe mesh that connects all clients to each other.
- Across three clouds, Clockwork outperforms the second best option in accuracy by 10x to 60x.

Accurately synchronized clocks will usher in a wave of innovation in distributed system observability, network congestion control, and distributed consensus algorithms. With more accurate clock sync, tracing tools like Datadog show distributed events in the factual order they occurred, network congestion can be counteracted effectively by using one-way delay signals, consensus algorithms like Paxos reach agreement more quickly, and distributed databases like CockroachDB operate faster.

© 2026 CLOCKWORK. All rights reserved.

Reliable clocksync means increased system performance, better user experience, reduced cloud cost, higher system efficiency and lower energy usage. But how accurate is accurate enough? In this blogpost, we'll quantify the accuracy of clock sync systems that are currently available.

We need a solution that can be deployed anywhere, including in virtual machines on the cloud, so we rule out candidate solutions that require support by the network infrastructure (such as PTP), or require special hardware on the client side. The conventional solution for clock synchronization is the Network Time Protocol (NTP), which is standardized in RFC 5905. NTP uses a client-server architecture where the client exchanges probe packets with one or more nearby servers to transfer the current time from the server clock to the client clock while compensating for network packet transmission latency. NTP clients run on virtually every modern computer, including internet servers, virtual machines in the cloud, PCs and laptops, smartphones, and many consumer devices.

Clockwork's high-accuracy clock sync algorithm: The new kid on the block

Clockwork's high-accuracy time synchronization system is built on a fundamentally different architecture than NTP. Instead of following a client-server paradigm, it synchronizes an entire cluster of machines by forming a probe mesh. The clocksync agent on each machine periodically exchanges small UDP packets with five to ten other machines in the cluster, typically every 10 milliseconds. The agents determine the instantaneous clock offset and clock drift for each edge in the mesh. For every 2-second period, the agents send their estimates to the Clockwork coordinator, which refines them by performing a global optimization using the network effect. The coordinator determines the optimal corrections for each clock that minimizes the clock offsets, and returns the corrections to the agents. The agent on each VM updates a multiplier in the operating system kernel, slightly speeding up or slowing down the clock as prescribed by the corrections. (This is the same

© 2026 CLOCKWORK. All rights reserved.

interface that ntpd and chrony use to correct clocks.) All userspace processes benefit from the now-corrected system clock.

Compared to NTP, the Clockwork algorithm achieves better synchronization quality for two reasons:

- Clockwork uses a larger number of raw probe measurements to identify and compensate intermittent queuing delays and path noise. A support vector machine (SVM) algorithm selects the most reliable measurements while discarding the unreliable ones. NTP is designed to be overly frugal by today's network bandwidth standards, and has to make do with much fewer measurements with no way to denoise.

- Clockwork takes advantage of loops in the probe mesh to counteract path asymmetries, where a probe packet sent from A to B systematically takes longer than the reverse packet sent from B to A. This exploits a network effect: A larger number of nodes in the network means that asymmetries can be corrected more effectively. In contrast, asymmetry is fundamentally undetectable in meshless systems like NTP: on a single network link, path asymmetry looks exactly like a clock offset between the two nodes.

How much do these algorithmic improvements matter in practice? As it turns out, they make a big difference. In the following, we'll quantify the difference in detail, based on experiments on Google Cloud, and Amazon Web Services (AWS), and Microsoft Azure.

Quantifying clock sync quality: Experimental setup

© 2026 CLOCKWORK. All rights reserved.

Each experiment consists of an example cluster with at least 20 VMs, hosted by one of three cloud providers, and running for at least 48 hours. In all three clouds, the test clusters are located in the US-East (Virginia) region. Each test cluster runs one of three candidate clocksync systems under test, for the duration of its lifetime. There were a total of nine experiment runs (three clouds times three clocksync systems).

As benchmark clocksync systems under test, we use two implementations of NTP: First, the reference implementation ntpd, which is the default on many Linux distributions. Second, a more recent implementation chrony, which uses the same network protocol as ntpd, but uses more advanced filtering and tracking algorithms to maintain tighter synchronization. Chrony is widely considered the best available NTP implementation (claimed by the authors and independently verified), both in terms of features and performance. All cloud providers now recommend chrony, and suggest an optimized configuration file that is also included in the default VM images. The recommended settings differ significantly between clouds, we describe them below. For both ntpd and chrony, each VM in the system exchanges packets with the configured NTP time server, but not with each other.

The third clocksync system under test is the most current release of the Clockwork's software.

For all three clocksync systems under test, we measure the clock sync quality by running a separate copy of the Clockwork software in observation mode. This observation system exchanges its own probe packets to determine the relative synchronization between the clocks, but it operates in read-only mode and does not control the clocks. It's important to stress that the observing system is completely independent of the primary systems. This guarantees that the data points used by the observing system are independent of the measurements used by the clock-controlling system. In machine learning terms, we ensure that the validation data used by the observing system is independent of the training data used by the controlling system.

© 2026 CLOCKWORK. All rights reserved.

In all cases, we take the clock offset between any two VMs as the metric by which to measure clock sync accuracy. We measure reliability by computing the probability that a certain target accuracy is met. Clearly, we target very high reliability values, so that systems that depend on clock sync to function properly will have a low probability of outage.

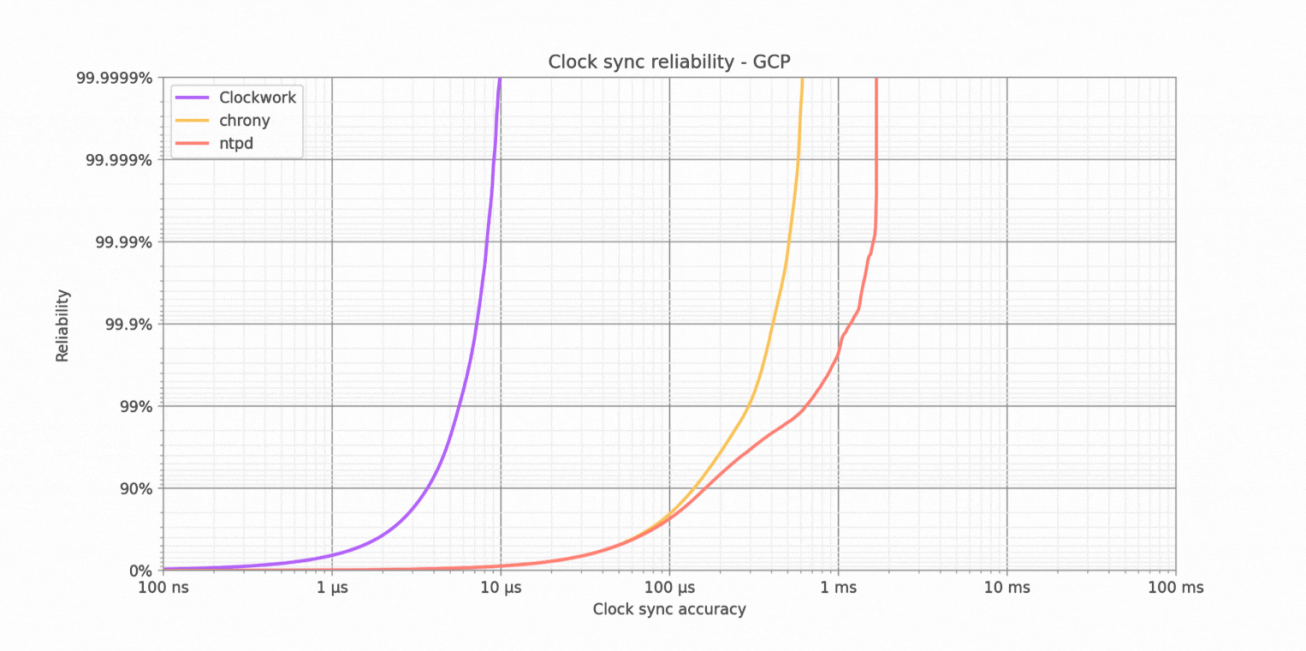
During the experiments, we do not run any load on the network. We expect the Clockwork system to have an added advantage if network load were added, since its denser probing allows it to detect and filter out the extra latencies caused by network load. An exact quantification of this expectation is left for a future blog post.

Without further preparatory ado, let's jump into the results for each of the clouds.

Google Cloud Platform (GCP)

For the experiment on GCP, we used VMs of type **n1-standard-1**. Both ntpd and chrony are configured to retrieve time from the NTP server at **metadata.google.internal**, which automatically resolves to a nearby server for each VM.

© 2026 CLOCKWORK. All rights reserved.



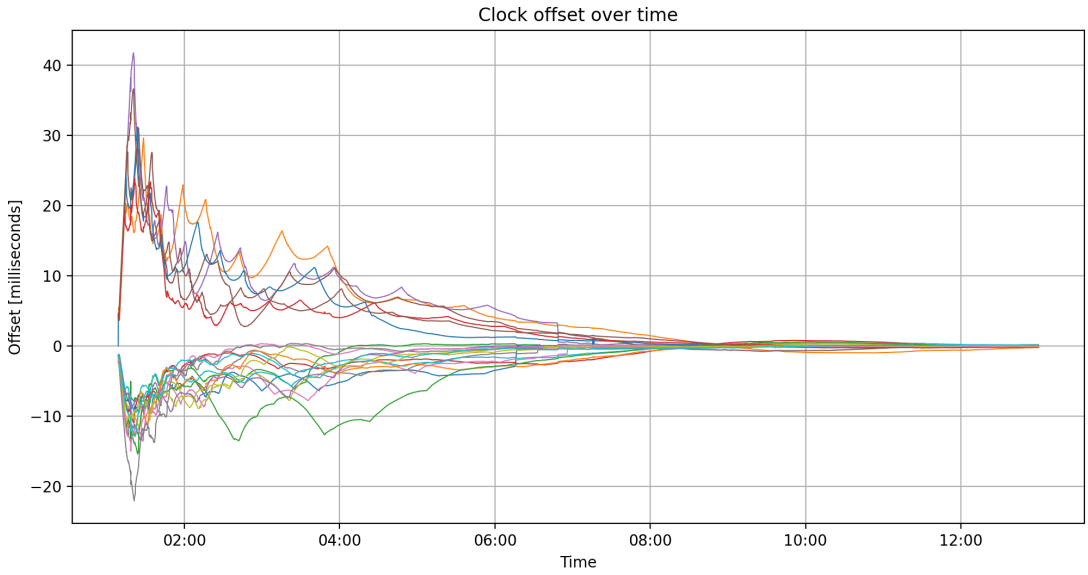
Clockwork's synchronization outperforms chrony's by a factor of 52x at 99% reliability (two nines), a factor of 62x at 99.99% probability (four nines), and a factor of 62x at 99.9999% (six nines). It also outperforms ntpd by a factor of 100x or better at all these benchmark points.

In absolute numbers, Clockwork achieves clock sync accuracy better than 10µs under all practical conditions. This exceeds the strictest requirements for clocksync in the financial industry hands-down, while neither chrony nor ntpd are able to do so (for example, the European MiFID II requirement for high-frequency trading is 100µs). It even beats the performance reported for PTP on physical machines (without virtualization), where previous studies report an accuracy of 30µs. Note that all results reported here are using software timestamps, since they are widely available in the cloud. When hardware timestamps are available, clocksync accuracy can be further improved by several orders of magnitude (see published results).

The graph above is created after each of the clocksync algorithms achieves convergence to steady-state condition. In the case of ntpd, this can take approximately 8 hours. In other words, the ntpd

© 2026 CLOCKWORK. All rights reserved.

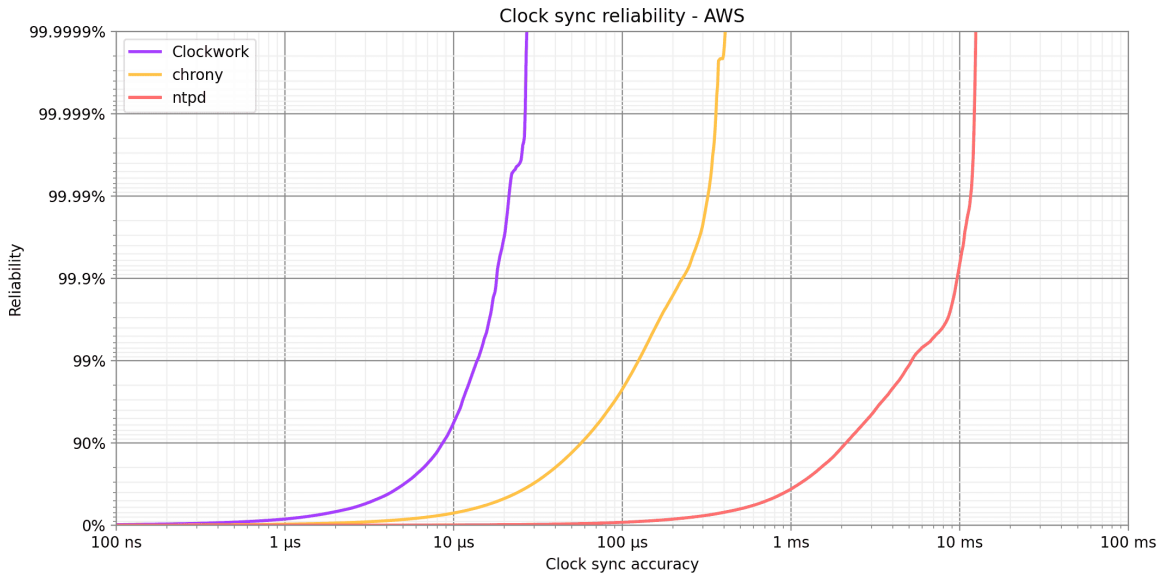
clock sync performance could be much worse than indicated in the plot above, especially for short-lived VMs where it is not possible to wait for convergence. When VMs are temporarily suspended or undergo live migration, ntpd may take several hours to reach convergence again. Both Clockwork and chrony achieve convergence much faster, typically within a couple of minutes.



Amazon Web Services (AWS)

For our AWS experiments, we use EC2 instances of type **m5.large**. ntpd was configured to use the server pool provided by Debian (**debian.pool.ntp.org**). For chrony, we follow the AWS recommendation to use the AWS-provided time NTP server at **169.254.169.123**, which resolves to a physical time server close to the VM.

© 2026 CLOCKWORK. All rights reserved.



At a high level, the comparison between the three clock sync solutions is similar to GCP: Clockwork performs better than chrony, which in turn performs better than ntpd. Compared to the GCP results, Clockwork's performance is a little worse (though still beating the 100µs goalpost with flying colors), while chrony's performance is a little better. The ntpd performance is much worse. This is because ntpd here uses time servers that are in the open internet rather than local to the cloud, as in the GCP case.

Numerically, Clockwork's synchronization outperforms chrony's by a factor of 9x at the 99% reliability level, a factor of 15x at the 99.99% reliability level, and a factor of 15x at the 99.9999% reliability level. Clockwork outperforms ntpd by a factor more than 200 at all levels.

Microsoft Azure

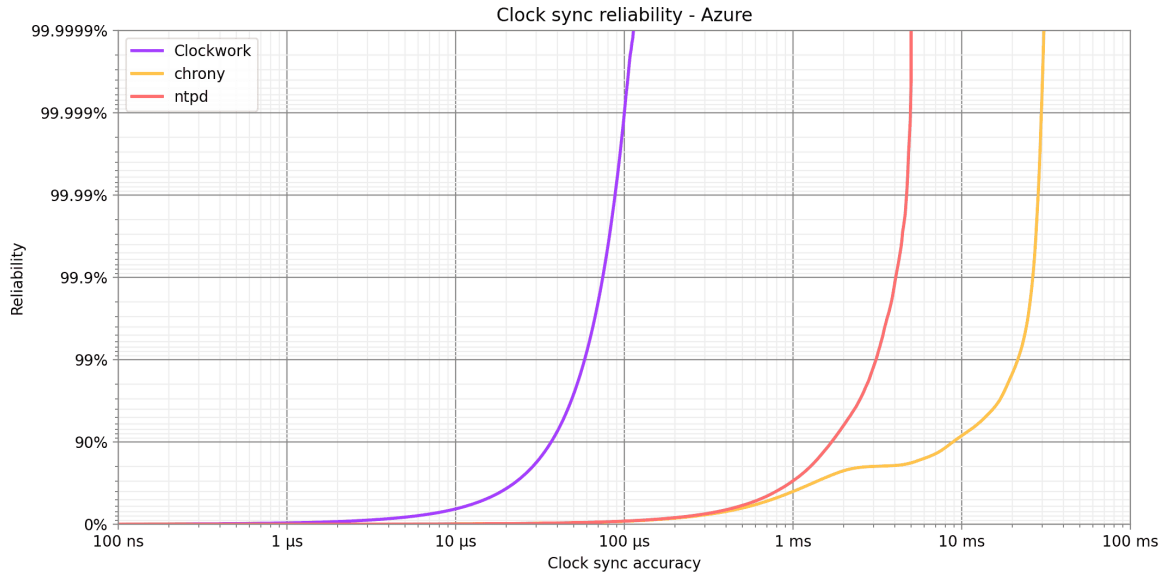
For our Azure experiments, we use VM instances of type **Standard_D1_v2** for ntpd and Clockwork, and **Standard_DS1_v2** for chrony. In both cases, accelerated networking was not turned on. ntpd

© 2026 CLOCKWORK. All rights reserved.

was configured to use the server pool provided by Debian (**debian.pool.ntp.org**), similarly to the AWS experiments. For chrony, we use the configuration that is present in the Debian VM images provided by Azure. This configuration does not configure NTP servers, but instead instructs chrony to use a **PTP hardware clock (PHC)**:

Azure hosts are synchronized to internal Microsoft time servers # that take their time from Microsoft-owned Stratum 1 devices. # The Hyper-V drivers surface this time source as a PTP-based # time source in the guest. This configures chrony to use it. refclock PHC /dev/ptp_hyperv poll 3 dpoll -2 offset 0

This ties the VM's system clock to the clock of the physical hosts on which it runs, which in turn is managed by Azure. As such, the way chrony is used in Azure is different from the GCP and AWS cases: in Azure, chrony does not actually talk to time servers via the NTP protocol. It is still a fair comparison, since in all three cases, the chrony experiment follows each cloud provider's recommendation on how to best supply VMs with accurate time.



© 2026 CLOCKWORK. All rights reserved.

Surprisingly, chrony performs worse than ntpd, even though ntpd is using time servers on the open internet. We cannot pinpoint the root cause, since chrony is relying on the underlying physical machine's clock, and we have no visibility into the clock sync process there. Chrony self-reports tight synchronization within tens of microseconds to its time source, which hints towards the time source itself not being synchronized well between VMs. The ntpd results are very similar to the ntpd results in AWS with the same time servers.

Clockwork's synchronization outperforms ntpd by a factor of 53x at the 99% reliability level, a factor of 54x at the 99.99% reliability level, and a factor of 44x at the 99.9999% reliability level. Interestingly, Clockwork's sync quality is worse here than in AWS or GCP, which could likely be alleviated by turning on Azure's accelerated networking to enable more accurate timestamp information.

Clockwork's time synchronization algorithm is a game changer

Across all three clouds, the Clockwork solution performs better than the second-best solution by an order of magnitude or more. Clockwork's architecture is mesh-based and exchanges probe packets more frequently and across many more links. Nevertheless, it is a lightweight solution in terms of resource usage: The clock sync agent requires less than 1% of a single core CPU, and produces sustained network traffic at rate of 700kbps, less than 0.04% of a 2Gbps link (the slowest link capacity you can select in today's clouds).

Curious about how Clockwork's solution can help your distributed systems perform better? Clockwork's clock sync is simple to deploy. We offer an enterprise solution and a developer version.

© 2026 CLOCKWORK. All rights reserved.

For enterprises, Clockwork's clock sync software leverages your existing time source to synchronize all your business clocks no matter your environment — on-prem, in the cloud, or a hybrid. Contact our sales team to learn more and schedule a demo.

For developers, Clockwork's clock sync is available as a free API providing time-as-a-service to synchronize your clocks using our time source. Sign up for early access here.

Interested in solving challenging engineering problems and building the platform that powers the next generation time-sensitive application? Join our world-class engineering team.

© 2026 CLOCKWORK. All rights reserved.