



GPU Cluster Time Synchronization: NTP vs PTP Acceptance Matrix

Plan and build private AI compute with GPU Smith. We connect workload requirements with GPU hardware, networking, deployment and operating decisions so your infrastructure fits the work it must perform.

Published by GPU Smith · 2026-10-31 · gpu cluster time synchronization · ntp vs ptp · precision time protocol · chrony vs linuxptp · ptp grandmaster · nvidia nvsm · nvidia firefly · spectrum-x ptp · clock drift · air-gapped time sync

Original article: <https://gpusmith.com/articles/en/gpu-cluster-time-synchronization-ntp-ptp>



In this article

1. Executive Summary
 2. Introduction and Background
 3. Time-Domain Map: Which Clock Matters Where
 4. NTP vs PTP for GPU Clusters: What Each Protocol Delivers
 5. Precision Time Protocol in AI Data Center Infrastructure: NVIDIA Reference Design, Versions and Questions to Ask
 6. Source Failure, Holdover and Failover
 7. Data Analysis and Evidence
 8. Implementation: Evidence, Acceptance Tests and Records
 9. Implications and Future Directions
 10. Frequently Asked Questions (FAQs)
 11. Conclusion
-

Executive Summary

Choosing between the **Network Time Protocol (NTP)** and the **Precision Time Protocol (PTP)** for a GPU cluster is a per-role decision, not a cluster-wide one. The sources reviewed for this report as of October 2026 support a split design: **chrony-class NTP** for control-plane, management, storage and monitoring clocks, and **a measured PTP domain** only where a named trace, telemetry or application requirement cannot be met by NTP. The RFC for NTP version 4 describes typical clients on fast local networks as within **a few hundred microseconds** (www.rfc-editor.org), and Red Hat documents tens of microseconds on a LAN (docs.redhat.com). Documented tolerances in common cluster software are far looser: **Ceph** Monitors warn at **0.05 seconds** of drift (docs.ceph.com), **Kerberos** defaults to **300 seconds** (web.mit.edu), and **MUNGE** credentials live five minutes by default (github.com).

The concrete example that motivates this report is documented by NVIDIA. In **NVIDIA System Management (NVSM)** multinode mode, if compute node and aggregator time are not synchronized, compute node MQTT messages are assumed stale (docs.nvidia.com) and NVSM CLI commands error out (docs.nvidia.com). The guide requires NTP-synced clocks (docs.nvidia.com) and states **no numeric skew threshold**, so the margin must be measured on the buyer's own aggregator and nodes. That is an acceptance test for the time source, not an argument for PTP.

PTP is justified by specific requirements. NVIDIA's **Firefly** reference design contrasts millisecond-level NTP (docs.nvidia.com) with applications that need **tens of nanoseconds to microseconds** (networking-docs.nvidia.com), and it keeps control-plane nodes on NTP (networking-docs.nvidia.com). NVIDIA's **Nsight Systems** documentation gives NTP-based multi-report alignment errors of **one to tens of milliseconds** (docs.nvidia.com) and recommends PTP-class synchronization for detailed distributed analysis (docs.nvidia.com). Neither source claims that every AI workload needs sub-microsecond time, and no source reviewed claims PTP improves GPU compute correctness.

Failover and **holdover** decide whether timestamps stay usable. Grandmaster datasheets quote 24-hour holdover drift from hundreds of microseconds for a standard crystal down to under one microsecond for rubidium (ww1.microchip.com), while PTP carries a timescale that differs from UTC by **37 seconds**

(hpiers.obspm.fr) (linuxptp.nwtime.org). The report therefore specifies an acceptance matrix for offset, drift, source loss, re-lock and log ordering, plus a worst-case relative timestamp bound equal to the sum of two absolute offset bounds. That bound is arithmetic, not measured accuracy. The decision rule: write down the maximum permitted skew from your own application and statement of work, choose the simplest protocol that meets it with margin, and prove it with evidence from normal operation and from a deliberate time-source failure.

Introduction and Background

This report is an analyst review of public documentation, not a benchmark. It contains no measurements taken by the publisher, and it makes no claims about any specific customer deployment. Every figure below is attributed to the document that states it, and where a document states no number, the report says so.

The audience is enterprise GPU cluster operators, network engineers, observability owners and air-gapped platform teams. Their shared question is practical: for each class of clock in a cluster (a head node, a baseboard management controller, a host operating system, a data processing unit, a switch), should the clock follow NTP or PTP, and what evidence shows it is good enough? "Good enough" is deliberately not defined by a universal threshold here. The buyer's application and statement of work (SOW) define the maximum permitted skew, and the purchased fabric defines what is physically achievable.

Three ideas anchor the analysis:

- **Several clocks per server.** NVIDIA's DGX service documentation describes a hardware real-time clock that the operator sets with `hwclock -w` after setting the date when NTP is not used (docs.nvidia.com).
- **Different timescales.** NTP's era 0 is defined in UTC from 1 January 1900 (www.rfc-editor.org), whereas the PTP epoch is 1 January 1970 00:00:00 TAI (www.rfc-editor.org).
- **Path dependence.** A protocol is only as good as its path: Red Hat advises PTP-enabled hardware between PTP clocks for best accuracy (docs.redhat.com).

The report proceeds in order: clock-dependent failure modes, time sources and trust boundaries (connected and air-gapped), NTP and PTP domains by node role, evidence for offset, drift, holdover and failover, acceptance checks for NVSM and cross-layer correlation, and an alert, change and incident record.

Time-Domain Map: Which Clock Matters Where

The documented NVSM stale-message behaviour

NVSM multinode is a useful anchor because NVIDIA documents both the dependency and the symptom. The aggregator is the central coordinator for the compute nodes. The guide states why time matters: since the nodes and aggregator connect over MQTT, the timestamp on MQTT messages needs to be in sync (docs.nvidia.com). It names the service, "ntptime or **chrony**", as acceptable (docs.nvidia.com). If synchronization is missing, the compute node MQTT messages are assumed stale (docs.nvidia.com).

Version and prerequisite facts, as of the 26.06 guide:

- **Compute-node version:** NVSM multinode supported version **26.03.01** onward (docs.nvidia.com), whereas the 25.03 guide stated **24.03.05** (docs.nvidia.com).
- **BCM guidance:** the **Base Command Manager (BCM)** provisioning page says multinode works on the older version but recommends 26.03.01 or above (docs.nvidia.com).
- **Aggregator placement:** on BCM-managed DGX SuperPOD deployments the aggregator runs on the BCM head node (docs.nvidia.com), so the head node's clock is the aggregator's clock.
- **Telemetry:** out-of-band telemetry is based on BMC Redfish URIs (docs.nvidia.com) and is stored with timestamps for historical analysis (docs.nvidia.com), so the aggregator is where host-clock and BMC-clock domains meet.

The guide gives no staleness window in seconds, so the tolerated skew is unknown until tested, and it describes NTP-class synchronization only.

Six clock domains and their trust boundaries

Table 1 below maps the clocks in a typical NVIDIA-based GPU cluster to a protocol choice, an upstream source, a failure behaviour to document and an owner. The protocol column is a recommendation derived from the cited documents; the "owner" column is a suggested accountability assignment, not a vendor statement.

Clock domain	Suggested protocol	Upstream source and documented behaviour	Failure behaviour to document	Suggested owner
Control plane (head node, NVSM aggregator, Kubernetes control plane)	NTP (chrony)	NVIDIA's reference design keeps control-plane nodes on NTP.	Aggregator and nodes drift apart: NVSM messages are assumed stale.	Platform team
BMC (out-of-band)	NTP	The DGX A100 BMC firmware release notes record NTP sync (docs.nvidia.com).	BMC and host timestamps differ by a zone or offset.	Infrastructure or hardware team
Host operating system (compute node)	NTP by default; PTP-disciplined only if the workload demands it	In the Firefly design the host system clock is synchronized from the DPU's hardware clock (networking-docs.nvidia.com), and NTP must then be disabled on worker nodes (networking-docs.nvidia.com).	Two writers (NTP and phc2sys) fight over the system clock.	Platform and OS team
DPU (BlueField)	PTP via Firefly	The DPU runs a PTP client that synchronizes hardware clocks from the boundary clock (networking-docs.nvidia.com).	Loss of PTP lock leaves the DPU on a free-running hardware clock.	Network team
Ethernet fabric (Spectrum switches)	PTP boundary clock, NTP for management	Cumulus Linux supports boundary clock mode only (docs.nvidia.com); PTP and NTP cannot both set the switch system clock (docs.nvidia.com).	Switch loses upstream; downstream DPUs lose the reference.	Network team

Clock domain	Suggested protocol	Upstream source and documented behaviour	Failure behaviour to document	Suggested owner
InfiniBand switches (NVOS)	NTP (as documented)	NVOS documentation describes the system clock for timestamping messages (networking-docs.nvidia.com) and enables NTP over DHCP by default (networking-docs.nvidia.com); no PTP section was found in the pages reviewed.	Switch event logs offset from host logs.	Network or fabric team

Table 1 is a starting design, not a standard: NTP governs the management plane, and PTP, where used, is limited to the data-plane path with **hardware timestamping** from **grandmaster** to host. Mixed NTP-and-PTP is the likeliest source of hidden error, which is why Firefly and Cumulus both document that the two protocols must not both discipline the same system clock (docs.nvidia.com).

Host, DPU and BMC clock relationships

The relationships between clocks are documented only partially. In the Firefly reference architecture, a host-side virtual function shares the DPU's PTP hardware clock ([networking-docs.nvidia.com](https://docs.nvidia.com/networking-docs/)), and the PHC runs on the TAI timescale while system time is UTC, so the validation procedure expects a difference of 37 seconds ([networking-docs.nvidia.com](https://docs.nvidia.com/networking-docs/)). By contrast, no NVIDIA document reviewed describes a PTP-disciplined BMC clock for DGX systems; the BMC clock evidence found is NTP-based and firmware-specific, and some of it concerns the older DGX A100. The relationship between a BMC clock and the host clock should therefore be treated as an acceptance-test item for the exact purchased platform, not an assumption.

NTP vs PTP for GPU Clusters: What Each Protocol Delivers

Accuracy claims, with their originators

The NTP specification gives modest typical numbers. RFC 5905 says typical primary servers on modern machines are precise within a few tens of microseconds (www.rfc-editor.org), secondary servers and clients on fast LANs within a few hundred microseconds (www.rfc-editor.org), and wide-area clients within a few tens of milliseconds (www.rfc-editor.org). Red Hat's RHEL 9 documentation states typical Internet accuracy of a few milliseconds (docs.redhat.com) and LAN accuracy of tens of microseconds (docs.redhat.com), adds that NTP hardware timestamping may reach sub-microsecond accuracy under ideal conditions (docs.redhat.com), and says PTP relies on assistance in network switches and routers (docs.redhat.com), while NTP hardware timestamping requires a polling interval shorter than the default range (docs.redhat.com).

For PTP, the IEEE 1588-2019 abstract says the standard supports synchronization accuracy and precision in the sub-microsecond range with minimal network and computing resources (standards.ieee.org), and NVIDIA's switch documentation repeats "sub-microsecond" as a capability (docs.nvidia.com). None of these statements is a guarantee for a particular site; NVIDIA's Cumulus documentation warns that 1G links may have lower PTP accuracy because of hardware limitations (docs.nvidia.com).

NTP vs PTP for GPU clusters: the decision rule

Use the following decision rule per node role. Each criterion must be tied to a written requirement.

- **NTP suffices** when the strictest consumer is a management service, a scheduler or a log pipeline with a documented tolerance measured in tens of milliseconds or more. **Slurm** requires synchronized clocks across the cluster (slurm.schedmd.com), and MUNGE needs the fastest and slowest clocks in the realm to fit within the credential time-to-live (github.com). Ceph requires NTP or PTP daemons on Monitor hosts (docs.ceph.com) and advises running NTP servers on bare metal, because virtualized clocks are unsuitable (docs.ceph.com).
- **PTP is justified** when a named trace or application requirement needs finer relative ordering than NTP can evidence. Examples with sources: the Nsight Systems guide's Ethernet switch metric sampling requirements state that PTP is strongly recommended, especially when correlating samples from multiple systems (docs.nvidia.com).
- **A third option exists** when PTP is unavailable: the Nsight Systems **vClock** plugin measures relative offsets between system clocks and targets correlation error on the order of tens of microseconds (docs.nvidia.com).

The sources do not support the claim that all distributed training needs sub-microsecond time. They support a narrower statement: if your acceptance criteria include cross-node timeline correlation finer than the NTP error you can evidence, you need PTP, a vClock-style relative measurement, or both.

Chrony vs linuxptp for a GPU cluster

The two tools are complementary, not rivals. Chrony is an NTP implementation; **linuxptp** is a PTP stack whose **ptp4l** daemon implements boundary, ordinary and transparent clocks (linuxptp.nwtime.org) and defaults to hardware timestamping (linuxptp.nwtime.org) and to the end-to-end delay mechanism (linuxptp.nwtime.org). Hardware timestamping needs NIC support: chrony's `hwtimestamp` directive needs Linux 3.19 or newer and a NIC that supports hardware timestamping (chrony-project.org), and NIC capabilities are checked with `ethtool -T` (networking-docs.nvidia.com). The linuxptp helper **phc2sys** is typically used to synchronize the system clock to a PTP hardware clock (linuxptp.nwtime.org), and Red Hat notes it is needed only with hardware time stamping (docs.redhat.com).

Two integration points matter operationally. First, because the PHC follows the PTP timescale while the system clock follows UTC, **phc2sys** maintains the offset between them (linuxptp.nwtime.org), and a wrong offset can leave the system clock wrong (linuxptp.nwtime.org). The **ptp4l** default for the TAI-UTC offset is 37 (linuxptp.nwtime.org), matching the IERS statement that UTC-TAI has been -37 seconds since 1 January 2017 (hpiers.obspm.fr). Second, chrony can consume a PHC as a reference clock, and its documentation says a PHC that runs on TAI (for example one disciplined by a PTP daemon) needs its UTC-TAI offset configured (chrony-project.org). Record which daemon writes to the system clock, because two writers produce conflicting corrections.

Chrony's own tracking output provides the evidence for NTP nodes: "System time" is the current offset between the NTP clock and the system clock (chrony-project.org), and "Last offset" is the estimated local offset at the last clock update (chrony-project.org). For **ptp4l**, summary statistics are reported in nanoseconds and parts per billion (linuxptp.nwtime.org).

Leap seconds and UTC behaviour

NTP carries leap warnings in-band: a leap indicator value of 1 means the last minute of the day has 61 seconds (www.rfc-editor.org). PTP's timescale is continuous and shifted against UTC by tens of seconds (linuxptp.nwtime.org). **IERS Bulletin C 72**, dated 6 July 2026, announced that no leap second will be introduced at the end of December 2026 (hpiers.obspm.fr), and the 2022 General Conference on Weights and Measures resolution decided to increase the maximum UT1-UTC difference in or before 2035 (www.bipm.org). Appliances still differ in timescale handling. Chrony offers a `leapsecmode` choice, in which the default system mode lets the kernel step the clock backwards when inserting a leap second (chrony-project.org), and for sources that do not announce leaps a timezone with leap data must be updated at least 12 hours ahead (chrony-project.org). Amazon documents that its PTP hardware clock offers no smeared time option (docs.aws.amazon.com), so mixing smeared and non-smeared sources needs an explicit policy; its local hardware clock also requires ENA driver version 2.10.0 or later (docs.aws.amazon.com). Leap-second behaviour of a specific grandmaster, switch or BMC must come from that appliance's documentation and be tested.

Precision Time Protocol in AI Data Center Infrastructure: NVIDIA Reference Design, Versions and Questions to Ask

Reference topology: PTP configuration for an AI training cluster

NVIDIA's reference design guide (RDG) for **DOCA Platform Framework** with Firefly builds a hierarchy: a grandmaster, an SN3700 switch acting as PTP **boundary clock** (networking-docs.nvidia.com), and **BlueField-3** DPUs running Firefly as PTP clients that synchronize the DPU hardware clocks. The host clock is then disciplined from the DPU. The profile aligns closely with the ITU-T G.8275.1 telecom profile (networking-docs.nvidia.com), and the switch boundary clock is configured to meet G.8273.2 Class C requirements (networking-docs.nvidia.com). The Firefly service supports two trusted-function interfaces for PTP link redundancy (networking-docs.nvidia.com), and the DPF Firefly reference includes a G.8275.1 profile configuration that users may adjust (github.com).

The reference design has stated limits that matter for procurement. Its grandmaster is a Linux server used for demonstration, and the RDG says it may not meet formal PTP grandmaster clock performance standards (networking-docs.nvidia.com). The RDG's validation monitor reports root mean square offset from the master in nanoseconds (networking-docs.nvidia.com), but the page text reviewed gives no numeric pass threshold, so the acceptance threshold remains the buyer's.

Versions and prerequisites, as of October 2026

- **Firefly guide (DOCA):** firmware must be **24.34.1002** or higher (docs.nvidia.com), and supported BlueField image versions are **3.9.0** and higher (docs.nvidia.com). Always check the version-specific guide for your DOCA release.
- **DPF integration:** the DPU flavor disables NTP on the DPU because Firefly needs a functional PTP setup (github.com).

- **Cumulus Linux (documentation for 5.18):** the documentation describes hardware timestamping of PTP packets (docs.nvidia.com), end-to-end delay only (docs.nvidia.com), and a single PTP domain per network (docs.nvidia.com).
- **Onyx:** NTP must be disabled before PTP is enabled (networking-docs.nvidia.com), and PTP is documented as unsupported on Spectrum-3 based switches (networking-docs.nvidia.com).
- **NIC and DPU hardware:** the **ConnectX-7** datasheet lists G.8273.2 Class C compliance (nvidia.com). Time stamping is off by default and must be enabled before use (networking-docs.nvidia.com).

A change-control caution applies to switches. Enabling the PTP service through the NVUE command line makes the platform restart switchd, which resets all network ports (docs.nvidia.com), so enabling PTP is a maintenance-window change, not a live toggle.

What Spectrum-X time synchronization does and does not tell you

The **Spectrum-X** question is common in search queries, so the evidence is stated plainly. The PTP documentation is generic to Spectrum switches running **Cumulus Linux**. NVIDIA **NetQ** monitors PTP on Spectrum switches running Cumulus Linux 5.0.0 and later (docs.nvidia.com). No Spectrum-X-specific accuracy figure was found, and no per-ASIC accuracy table exists on the Cumulus page. The consequence for buyers is direct: whether hardware timestamping and boundary-clock support exist in the exact purchased fabric must be confirmed from the vendor's documentation for that switch model, operating system release and NIC firmware, then proven by measurement.

Hardware details that affect timestamps

Enabling time stamping on ConnectX-4 and later adapters disables receive completion-queue compression (networking-docs.nvidia.com). ConnectX-6 Dx and later devices default to free-running internal time mode (networking-docs.nvidia.com), only physical functions can modify the hardware real-time clock (networking-docs.nvidia.com), and all functions on one device must sync to the same 1588 master (networking-docs.nvidia.com). Each point bears on how a DPU-hosted or SR-IOV design is validated.

Source Failure, Holdover and Failover

Grandmaster choices for a GPU cluster

A grandmaster is the PTP time source for the domain. NVIDIA's RDG states that its demonstration grandmaster may not meet formal grandmaster performance standards (networking-docs.nvidia.com).

Microchip's SyncServer S600 datasheet gives 24-hour holdover drift of **400 microseconds** with a standard oscillator, 25 microseconds with OCXO and under 1 microsecond with rubidium, evaluated after five days locked to GPS (ww1.microchip.com). **Meinberg** publishes an oscillator holdover table, and its LANTIME grandmasters state that the oscillator determines the holdover characteristics (www.meinbergglobal.com).

Safran's VersaSync datasheet offers a miniature rubidium option and can be re-synchronized from an alternative external reference (safran-navigation-timing.com). The Microchip S650 lists four gigabit ports with NTP and PTP hardware timestamping (www.microchip.com).

The Open Compute Project's Time Appliances Project describes the Time Card as GNSS-disciplined with a high-stability holdover oscillator to maintain accuracy if GNSS fails (github.com), and an Open Time Server Cluster workstream extends the Data Center PTP Profile to clusters of PTP grandmasters (www.opencompute.org). Meta describes a fleet design committed to supporting up to one million clients per appliance (engineering.fb.com), distributing TAI to clients (engineering.fb.com), and rehearsing holdover by disabling GNSS constellations on the Time Card receiver (engineering.fb.com); the transferable lesson is that failover was exercised deliberately. Meinberg lets operators configure the priority of every reference input (www.meinbergglobal.com) and delays falling back from GNSS until the internal oscillator can no longer beat the alternative (www.meinbergglobal.com), which matters when designing air-gapped fallback chains.

Best master clock behaviour and failover timing

Failover is decided by the **best master clock algorithm (BMCA)**. In `linuxptp` the number of missed Announce messages before the last Announce expires is `announceReceiptTimeout` (manpages.debian.org), default 3 (manpages.debian.org), and the clock class threshold at which a client leaves lock state defaults to 248 (manpages.debian.org). The `clockClass` option denotes traceability of the time distributed by the grandmaster (linuxptp.nwtime.org). A G.8275.1 summary notes that a grandmaster in holdover within specification advertises a distinct clock class from one with no reference (calnexsolutions.atlassian.net), and that the profile uses an alternate BMCA permitting multiple active grandmasters for redundancy (calnexsolutions.atlassian.net). For fixed topologies, `ptp4l` can skip the BMCA and use static roles (manpages.debian.org). At Meta, a time appliance that loses GNSS advertises a holdover clock class so that clients migrate away (engineering.fb.com), and its design increases the window of uncertainty at a compound rate based on oscillator stability (engineering.fb.com).

Cumulus supports only one PTP domain per network (docs.nvidia.com), so redundancy must live inside one domain, and failover time (announce timeout, re-selection and servo re-lock) must be measured with the actual switches and DPUs, not inferred from defaults.

Air-gapped sites and NTP fallback

Air-gapped sites have no public NTP pool or default internet reference. Cumulus Linux cannot learn NTP servers from DHCP (docs.nvidia.com), whereas NVOS enables NTP over DHCP by default. An air-gapped design must configure an internal reference explicitly, such as a GNSS-disciplined time server in a controlled location, or a rubidium-backed appliance that can be re-synchronized from an alternative reference (safran-navigation-timing.com). If Network Time Security (NTS) is used internally, its key establishment runs over **TCP port 4460** (www.rfc-editor.org), which a firewall policy must permit.

Chrony offers several controlled-degradation mechanisms. The `local` directive with `orphan` is intended for isolated networks where hosts must agree with each other even if they are not synchronized to real time (chrony-project.org); failover among orphan servers goes by reference ID order (chrony-project.org). Raising `minsources` can improve reliability (chrony-project.org), `maxdistance` rejects sources beyond a default root distance of 3 seconds (chrony-project.org), and `makestep` is normally for boot only (chrony-project.org). A tracking Reference ID that shows local mode means the host is not synchronised to any external source

(chrony-project.org), which is the state the alert should detect. The older ntpd has its own orphan mode, enabled only after a delay with no sources (www.eecis.udel.edu). An isolated cluster that agrees with itself is internally consistent but not traceable to UTC, and the record should say which condition applies.

Drift between corrections

When all references fail, clocks drift at the oscillator's frequency error. Chrony's maxclockerror assumes **1 ppm** by default (chrony-project.org). At 1 ppm, a 50 millisecond relative error (the Ceph monitor limit (docs.ceph.com)) accrues in **about 14 hours**, an arithmetic illustration and not a measurement of any server. A real server's rate error must be read from chronyc tracking, where Frequency is the rate error and Skew is the estimated error bound on it (chrony-project.org).

Data Analysis and Evidence

Documented tolerances and capabilities

Table 2 collects numbers that bound the requirement from above (what software tolerates) and from below (what each protocol or device claims). The entries are vendor or project documentation for each product's own facts. Several are defaults or typical values, not guarantees.

Item	Documented figure	What it means for the choice
NTP, fast LAN client (RFC 5905)	Within a few hundred microseconds (www.rfc-editor.org)	Adequate for most management-plane consumers.
NTP with hardware timestamping	Sub-microsecond possible under ideal conditions (docs.redhat.com)	Check before assuming PTP is needed.
PTP (IEEE 1588-2019 abstract)	Sub-microsecond range (standards.ieee.org)	Capability statement, not a site result.
Amazon EC2 clock error bound	Typically under 40 microseconds via PHC (aws.amazon.com); under 100 microseconds via NTP (aws.amazon.com)	A cloud-operator figure, not a cluster result.
Ceph Monitor clock drift	0.05 seconds default (docs.ceph.com)	A storage consensus tolerance.
Kerberos clock skew	300 seconds default (web.mit.edu)	A security tolerance.
MUNGE credential lifetime	5 minutes default (github.com)	Slurm authentication tolerance.
Nsight Systems, NTP-based multi-report alignment	One to tens of milliseconds (docs.nvidia.com)	Origin of the trace-level case for PTP.
Grandmaster 24-hour holdover (Microchip S600)	400 microseconds, 25 microseconds, under 1 microsecond by oscillator (ww1.microchip.com)	Sets how long a site can ride through GNSS loss.

Two observations follow. None of the cluster management tolerances approaches the sub-microsecond range, which supports NTP for management. The only entry that argues for PTP is the trace-alignment one: Nsight Systems' NTP-based error range of one to tens of milliseconds is wide enough that sub-millisecond ordering between nodes cannot be asserted from NTP alone.

The relative timestamp bound

Two events logged on different clocks can be ordered reliably only if their true separation exceeds the combined uncertainty of both timestamps. The simple arithmetic bound is:

relative timestamp uncertainty bound = |offset bound of clock A| + |offset bound of clock B|

This is an arithmetic bound derived from two assumed absolute offset bounds. It is not measured system accuracy, and it ignores timestamping point error (an application logging after a scheduling delay), queueing and clock steps.

(Hypothetical Example) If an audit of the NTP clients in a rack bounds clock A at 2 milliseconds and clock B at 3 milliseconds, the bound is 5 milliseconds, so two log lines closer than 5 milliseconds apart cannot be ordered with confidence. If the PTP-disciplined hosts' audited bounds were 1 microsecond each, the bound would be 2 microseconds. Both pairs of inputs are invented for the arithmetic and are not measurements.

Chrony publishes a related quantity: the clock error bound, built from system time offset, root dispersion and half the root delay. Amazon notes that with a direct PHC connection chrony will underestimate this bound, so Amazon adds the PHC's own error bound (aws.amazon.com). The lesson for cluster operators is that a daemon's self-reported error is a lower estimate of the true uncertainty whenever another clock hop sits beneath it, such as a DPU hardware clock feeding a host.

Implementation: Evidence, Acceptance Tests and Records

Collect offset, drift, holdover and failover evidence

Evidence has to cover normal operation and a time-source failure. The following inventory is the minimum, using only commands and fields the cited tools document.

- **NTP nodes:** record `chronyc tracking` System time and Last offset (chrony-project.org), source reachability, and whether the reference ID shows local mode (chrony-project.org).
- **PTP hosts:** record the `ptp4l` summary offset RMS and maximum offset, whose units are nanoseconds and parts per billion ([linuxptp.nwtime.org](https://linuxptp.org)), and the `phc2sys` offset between the PHC and `CLOCK_REALTIME`. Confirm the TAI-UTC offset matches the expected 37 seconds.
- **Switches:** collect the Cumulus PTP monitor counters and the last 25 PTP timestamps from the timestamp log (docs.nvidia.com), and use NetQ for fleet-level PTP views.
- **NICs and DPUs:** record `ethtool -T` output and the firmware and image versions against the Firefly prerequisites.
- **Observability pipeline:** record each exporter's sampling and scrape behaviour, so a sample's timestamp is not assumed to be the scrape instant.

No document reviewed prescribes a minimum collection window; the buyer must set one that spans normal load and a maintenance event, and record the reason.

GPU cluster clock synchronization acceptance testing: offset, drift, source loss, re-lock and log ordering

Table 3 is an acceptance matrix. The pass criteria are intentionally left as references to the buyer's own requirement, because no universal offset threshold exists in the sources and none should be invented. Vendor defaults appear only as reference points.

Test	Method	Evidence to retain	Pass criterion (set by buyer)
A1. Steady-state offset	Sample chrony offsets (chrony-project.org) or ptp4l offsets (linuxptp.nwttime.org) per node over the agreed window	Min, max, RMS offset per node; version of daemon and firmware	Offset bound within the SOW skew, with margin; record the bound, not only the average
A2. Drift	Disable upstream reachability on a lab node and read Frequency and Skew (chrony-project.org)	Frequency in ppm, time to cross the application tolerance	Time to tolerance exceeds the planned holdover
A3. Source loss (NTP)	Remove the primary reference; watch the tracking Reference ID (chrony-project.org) and failover order (chrony-project.org)	Time of loss, alert time, new reference, offset after failover	Alert fires; failover reference reachable; offset stays in bound
A4. Grandmaster loss (PTP)	Remove GNSS or the grandmaster; observe announce timeout (manpages.debian.org) and the clients' clock class threshold (manpages.debian.org)	Port states, grandmaster identity, offset trace through the event	No client exceeds the offset bound, or the exceedance is within the documented budget
A5. Re-lock	Restore the source	Time to lock, step events (chrony-project.org) or slew events, phc2sys behaviour (linuxptp.nwttime.org)	No unexpected clock step in application hosts
A6. NVSM staleness	In a lab, offset a compute node clock in increments while watching nvsm commands and MQTT state	Offset at which commands first return the error	Measured margin exceeds the maximum offset seen in A1 through A5
A7. Cross-layer log ordering	Generate a marker event that touches host, BMC, switch and application logs	Timestamps per layer; computed pairwise bound	Ordered within the bound in the relative-timestamp formula
A8. Timescale and leap handling	Compare PHC and system clocks; confirm the offset against the 37 second TAI-UTC value (hpiers.obspm.fr)	Offset readout; appliance leap setting	Offset equals expected; leap behaviour documented per appliance

Table 3 deliberately includes a test that offsets a lab node's clock to find the NVSM staleness margin. Because the NVSM guide states no threshold, this test converts an unknown into a number. It should run only in a lab or a maintenance window, never as a live experiment on a production aggregator.

Clock drift in distributed training logs: cross-layer correlation checks

Correlation fails quietly. The NVIDIA observability guidance lists a common timeline for GPU, node and fabric signals in one view as a selection criterion (developer.nvidia.com), and it names **DCGM** and NVSM **Prometheus** endpoints among the sources (developer.nvidia.com). The page states no timestamp rule, so the common timeline depends on the clock work in this report.

Run a marker test: emit a uniquely tagged event at a known moment on a host, and confirm the same tag appears with the expected ordering in the host log, the BMC event log, the switch log and the application trace. Then evaluate the downstream pipelines:

- **Metrics:** Prometheus accepts out-of-order samples only within a configured window (prometheus.io).
- **Logs: Loki** by default accepts entries only within a window of one hour behind the newest entry in a stream (grafana.com), and lists clock skew between log sources and Loki as something to check when ingestion errors occur (grafana.com).
- **Traces:** Jaeger's query service can adjust spans when a server span appears to start earlier than the client span (www.jaegertracing.io), which hides skew from the viewer instead of removing it.
- **Certificates:** X.509 validity is an interval, so a badly wrong node clock also breaks certificate checks (www.rfc-editor.org).

Operational alert, change and incident record template

The template below is a suggested structure. Bracketed items are fields to fill, not data.

```
TIME-SOURCE DRILL RECORD
Cluster / site: [name]           Date and window: [start - end]
Requirement source: [SOW clause / application doc]   Max permitted skew: [value, unit]
Roles under test: [control plane | BMC | host | DPU | switch | grandmaster]
Protocol and daemon versions: [chrony / linuxptp / Firefly / NOS / firmware]
Failure injected: [source removed | GNSS lost | link cut] at [time]
Alert fired at: [time]      Alert rule: [name]
Failover target: [source]   Failover completed at: [time]
Offset before / during / after: [bound] / [bound] / [bound]
Clock steps observed: [yes/no, where]
NVSM command state during event: [ok / error text]
Log ordering marker result: [pass/fail, pairwise bound]
Pass/fail vs requirement: [ ]   Owner sign-off: [name, role]
Follow-up changes: [ticket IDs]
```

Pair the drill record with an alert set:

- **Source loss:** loss of every external source, or a tracking Reference ID in local mode.
- **PTP state:** a port state change or a grandmaster identity change.
- **Offset:** any offset beyond the buyer's bound.
- **Clock steps:** any system-clock step.

Pair it with a change rule: switch PTP enablement is a disruptive change, and any change to which daemon writes the system clock needs a rollback plan because both Firefly and Cumulus document that NTP and PTP must not both set the clock.

Implications and Future Directions

Three implications follow for operators planning purchases and acceptance in 2026 and 2027.

- **Time becomes a bill-of-materials line.** Hardware timestamping and boundary-clock support are model-and-release questions: Cumulus documents boundary clock only, **Onyx** documents a Spectrum-3 limitation, and NIC behaviour depends on firmware and mode. Put these in the request for proposal beside the buyer's maximum permitted skew.
- **Monitoring stacks raise the cost of bad clocks.** NVIDIA publishes no skew threshold for the stale-message behaviour, so each operator should own a measured margin and re-measure after NVSM upgrades, since the minimum compute-node version moved from 24.03.05 to 26.03.01 between guide releases.
- **Leap-second policy is converging but unfinished.** With no leap second announced for the end of December 2026 (hpiers.obspm.fr) and an international resolution to widen UT1-UTC tolerance by 2035 (www.bipm.org), the operational risk is declining, but appliances still differ in timescale handling. Keep the A8 test in the acceptance matrix.

Open questions remain: no public benchmark in the sources measures achieved PTP offset on a Spectrum-X fabric end to end, and no NVIDIA document reviewed describes PTP for the DGX BMC. Operators should treat these as lines in their own test plans.

Frequently Asked Questions (FAQs)

Is NTP enough for a GPU cluster?

For management-plane and most telemetry use, the documented tolerances (0.05 seconds for Ceph (docs.ceph.com), 300 seconds for Kerberos (web.mit.edu)) are far looser than typical LAN NTP accuracy of a few hundred microseconds (www.rfc-editor.org). NTP is enough unless a trace or application requirement exceeds what you can evidence with it.

When is PTP justified?

When a written requirement needs cross-node ordering finer than NTP can evidence, for example multi-system trace correlation, where NVIDIA's profiler documentation recommends PTP. It is not justified by the claim that AI workloads in general need it, and the sources reviewed do not say PTP changes GPU compute correctness.

What does a PTP grandmaster for a GPU cluster need?

A GNSS-disciplined or atomic-referenced appliance with holdover figures that match your outage budget (compare datasheet holdover tables (ww1.microchip.com) (www.meinbergglobal.com)), PTP profile support that matches the fabric, and a documented air-gapped fallback (safran-navigation-timing.com).

How do I prove clocks stay usable during a failover?

Run drills A3 to A5 in Table 3, keep the drill record, and compare observed offsets (chrony (chrony-project.org), ptp4l (linuxptp.nwtime.org)) with the buyer's requirement, not with a vendor default.

Conclusion

The evidence supports a role-based answer to the question of GPU cluster time synchronization. Keep the control plane, BMCs, storage and monitoring on NTP with at least three independent references ([chrony FAQ](#)), confirmed by the A3 source-loss drill, and a documented air-gapped fallback. Reserve PTP for the data-plane path where a named requirement, such as multi-node trace alignment, justifies the added hardware, configuration and monitoring burden, and only after confirming that the purchased switches, NICs and DPUs provide hardware timestamping and boundary-clock support. NVIDIA's NVSM guide shows why even the NTP tier needs acceptance testing: documented stale-message behaviour with no published threshold.

The practical deliverables are an acceptance matrix with buyer-defined criteria, a source-loss drill with retained evidence, and an incident template that records the offset bound and any clock step. The worst-case relative timestamp bound (the sum of two absolute offset bounds) is a planning aid, not a measurement. Until those measurements exist for a given fabric, claims about achieved precision remain unknown, and an operator who has not measured them should not assert them.

Source links

Links cited in this article, in order of first appearance. Full addresses are provided for readers using extracted text.

1. <https://www.rfc-editor.org/rfc/rfc5905.html>
2. https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/9/html/configuring_basic_system_settings/configuring-time-synchronization_configuring-basic-system-settings
3. <https://docs.ceph.com/en/latest/rados/troubleshooting/troubleshooting-mon>
4. https://web.mit.edu/kerberos/krb5-latest/doc/admin/conf_files/krb5_conf.html
5. <https://github.com/dun/munge/wiki/Installation-Guide>
6. <https://docs.nvidia.com/datacenter/nvsm/nvsm-user-guide/26.06/nvsm-multinode.html>
7. <https://docs.nvidia.com/networking/display/public/SOL/RDG-for-DPF-Host-Trusted-with-Firefly-Time-Synchronization,-OVN-Kubernetes-and-HBN-Services>
8. <https://networking-docs.nvidia.com/sol/rdg-for-dpf-host-trusted-with-firefly-time-synchronization-ovn-kubernetes-and-hbn-services>
9. <https://docs.nvidia.com/nsight-systems/UserGuide/index.html>
10. <https://ww1.microchip.com/downloads/aemDocuments/documents/FTD/ProductDocuments/Brochures/SyncServer-S600-Data-Sheet-2903.pdf>

11. <https://hpiers.obspm.fr/iers/bul/bulc/bulletinc.dat>
12. <https://linuxptp.nwtime.org/documentation/phc2sys>
13. <https://docs.nvidia.com/dgx/dgxe100-service-manual/motherboard-tray-battery-replacement.html>
14. <https://www.rfc-editor.org/rfc/rfc9581.html>
15. https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/7/html/system_administrators_guide/ch-configuring_ptp_using_ptp4l
16. <https://docs.nvidia.com/datacenter/nvsm/nvsm-user-guide/25.03/nvsm-multinode.html>
17. <https://docs.nvidia.com/datacenter/nvsm/nvsm-user-guide/26.06/nvsm-multinode-bcm.html>
18. <https://docs.nvidia.com/dgx/dgxe100-fw-container-release-notes/dgxe100-fw-release-notes.html>
19. <https://docs.nvidia.com/networking-ethernet-software/cumulus-linux-518/System-Configuration/Date-and-Time/Precision-Time-Protocol-PTP>
20. <https://networking-docs.nvidia.com/nvosinfinibandum/25025019/time-synchronization>
21. <https://networking-docs.nvidia.com/nvosinfinibandum/25025019/ntp-commands>
22. <https://docs.nvidia.com/doca/sdk/doca-firefly-service-guide/index.html>
23. <https://standards.ieee.org/ieee/1588/6825>
24. https://slurm.schedmd.com/quickstart_admin.html
25. <https://docs.ceph.com/en/latest/rados/configuration/mon-config-ref>
26. <https://linuxptp.nwtime.org/documentation/ptp4l>
27. <https://chrony-project.org/doc/4.6/chrony.conf.html>
28. <https://networking-docs.nvidia.com/doca/sdk/time-stamping>
29. <https://chrony-project.org/doc/4.6/chronyc.html>
30. <https://www.bipm.org/en/cgpm-2022/resolution-4>
31. <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/set-time.html>
32. <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/configure-ec2-ntp.html>
33. <https://github.com/NVIDIA/doca-platform/blob/public-release-v25.10/docs/public/advanced-configuration/dpuserVICES/firefly.md>
34. <https://networking-docs.nvidia.com/onyxum/3102202/precision-time-protocol-ptp>
35. <https://nvidia.com/content/dam/en-zz/Solutions/networking/ethernet-adapters/connectx-7-datasheet-Final.pdf>
36. <https://docs.nvidia.com/networking-ethernet-software/cumulus-netq-412/Monitor-Operations/Monitor-PTP>
37. <https://www.meinbergglobal.com/english/products/lantime-m600-gps-ptpv2.htm>
38. https://safran-navigation-timing.com/wp-content/uploads/2023/02/SAFRAN_VersaSync-with-mRO-50-option_Datasheet_7-3-2024_US.pdf
39. <https://www.microchip.com/en-us/products/clock-and-timing/systems/gnss-timing-instruments/syncserver-s650>
40. <https://github.com/Time-Appliances-Project/Time-Card>
41. <https://www.opencompute.org/projects/time-appliances-project-tap/open-time-server-cluster>
42. <https://engineering.fb.com/2022/11/21/production-engineering/precision-time-protocol-at-meta>
43. <https://manpages.debian.org/testing/linuxptp/ptp4l.8.en.html>

44. <https://calnexsolutions.atlassian.net/wiki/spaces/GDW/pages/10485787/ITU-T+G.8275.1+PTP+Profile+-+FTS+One-Page+Summary>
45. <https://docs.nvidia.com/networking-ethernet-software/cumulus-linux-516/System-Configuration/Date-and-Time/Network-Time-Protocol-NTP>
46. <https://www.rfc-editor.org/rfc/rfc8915.html>
47. <https://chrony-project.org/doc/4.6.1/chrony.conf.html>
48. <https://www.eecis.udel.edu/%7Emills/ntp/html/orphan.html>
49. <https://aws.amazon.com/blogs/compute/its-about-time-microsecond-accurate-clocks-on-amazon-ec2-instances>
50. <https://developer.nvidia.com/blog/how-to-choose-full-stack-observability-for-nvidia-ai-factories>
51. <https://prometheus.io/docs/prometheus/latest/configuration/configuration>
52. <https://grafana.com/docs/loki/latest/operations/troubleshooting/troubleshoot-ingest>
53. <https://www.jaegertracing.io/docs/2.dev/deployment/configuration>
54. <https://www.rfc-editor.org/rfc/rfc5280.txt>
55. <https://www.meinbergglobal.com/english/specs/gpsopt.htm>
56. <https://chrony-project.org/faq.html>

THE PUBLISHER

About GPU Smith

GPU Smith provides independent engineering and research for private AI infrastructure. We help technical buyers and operators reason about GPU workload sizing, hardware selection, procurement, deployment and inference operations, from the compute system to the networking, power and cooling requirements around it.

Start with the workload

Useful infrastructure decisions begin with the models, data, throughput, latency and operating constraints a team actually has. GPU Smith helps connect those requirements with system design and deployment choices. The goal is a privately controlled AI environment whose capacity and operational demands are understood before a purchasing decision.

Hardware and supplier research

Our public reference library covers GPUs, complete systems and networking components. The server-vendor directory and manufacturing research help teams investigate suppliers, compare options and follow sources behind technical claims. We distinguish manufacturer specifications from measured benchmarks and advertised capabilities from tested configurations. Reference pages are research resources, not a live inventory listing or a binding equipment quote.

Deployment and operations

GPU Smith's engineering scope includes infrastructure integration, networking, power, cooling and the practical operation of inference workloads. Our published articles explain the tradeoffs behind deployment, procurement and ongoing operations so teams can ask better questions and document decisions.

Work with GPU Smith

Explore the [hardware library](#), [GPU references](#), [networking references](#) and [vendor research](#). [Contact GPU Smith](#) with your workloads and deployment constraints to discuss a project and confirm current scope, pricing and availability.

Inclusion of a manufacturer or product in our research does not imply a partnership, certification or endorsement.

Website: <https://gpusmith.com>

This article is educational. Verify consequential medical, legal, financial, regulatory and technical decisions with appropriate professionals and the cited primary sources. Company services are described separately from the article's research findings.