

What's the best way to synchronize times to millisecond accuracy AND precision between machines?

Asked 16 years, 3 months ago Modified 9 months ago

Viewed 45k times



39



From what I understand, the crystals on PCs are notorious for clock skew. If clocks are always skewing, what is the best way to synchronize clocks between machines, with millisecond accuracy and precision? From what I've found, NTP and PTP are possible solutions, but I was wondering if anyone else has had any experiences with PTP (IEEE1588).

sockets

synchronization

gps

time-management

Share

Improve this question

Follow

edited Feb 28 at 6:51



Community Bot

1 ● 1

asked Sep 18, 2008 at 23:05



PiNoYBoY82

1,648 ● 2 ● 14 ● 17

9 Answers

Sorted by:

Highest score (default)





Just run the standard NTP daemon.

19

It does have options to take input from several GPS devices as well as talking to network servers.



Edit: I was referring to <http://www.ntp.org/>, not the one that comes with Windows.



I don't have any suggestion as to what NTP clients are best for windows, but for Unix machines there's no real reason to not run NTP.

Share Improve this answer

edited Sep 18, 2008 at 23:33

Follow

answered Sep 18, 2008 at 23:08



wnoise

9,902 ● 39 ● 47



17

Here's some 15-year-old software that syncs to within a **hundredth** of a millisecond. (My team wrote it when NTP wasn't good enough for our lab.)



From the conference paper's [abstract](#): "A distributed clock for networked commodity PC's. With no extra hardware, this clock correlates sensor data from multiple PC's with latency and jitter under 10 microseconds average, 100 microseconds worst case."



Source code: <https://github.com/camilleg/clockkit>
(Until 2020 Feb 13 it was at
<http://zx81.isl.uiuc.edu/clockkit/>, now offline.)

Share Improve this answer

edited Apr 15, 2020 at 16:57

Follow

answered Mar 9, 2015 at 16:44



Camille Goudeseune

3,152 ● 2 ● 38 ● 57

I really wanted to follow these links but the zx81.isl.uiuc.edu links are dead and scholar.google.com links to zx81.isl.uiuc.edu/camilleg/dsceu04.pdf is dead as a result also :(– [gazarsgo](#) Oct 1, 2015 at 13:45

@gazarsgo try this: webcache.googleusercontent.com/...
– [AddingColor](#) May 3, 2017 at 13:10

Neato, @AddingColor, even if the diagrams didn't survive the conversion to HTML. Anyways, the [zx81](http://zx81.isl.uiuc.edu) links have remained up for a few years. – [Camille Goudeseune](#) May 3, 2017 at 16:12

@CamilleGoudeseune Would this work on the open internet? If so, to what degree of accuracy? (micro/nanoseconds)
– [rawinput](#) Dec 12, 2018 at 18:16

@CamilleGoudeseune How do i link with common c++ cause I installed libcommoncpp2-dev with apt (ubuntu 16) and imported headers normally but the linking is not working : undefined reference to `ost::IPV4Address::~IPV4Address()'
– [rawinput](#) Dec 12, 2018 at 20:59



9



You cannot synchronize machines to the level of milliseconds by exchanging data, because any data exchange itself already takes at least milliseconds to happen and thus spoils your result! Even protocols that try to first measure how long a data transfer takes and then sending out the time info (taking the measured delay into account) are just a bit better than average but they are still not good since not every data transfer takes equal time (just constantly ping a server on the Internet and see how every ping has a different delay).

The only way to really synchronize two computers in the milliseconds range is by having them both obtain the time from the same source via a transfer method that has no unknown or constantly changing delay. E.g. if both receive a satellite signal, that broadcasts the time. The signal will always have a constant delay (from satellite to earth) and they are both receiving it almost within the same nanosecond.

Germany for example has a radio controlled time. Somewhere in the country is an atomic clock (that has correct time to the nanosecond for hundreds of years) and some sender permanently broadcast the current time on a given frequency all over the country. Alarm clocks and even wristwatches exist that can receive this time and permanently synchronize with it (well, not really permanently, most models do that only once every 24 hours to save battery runtime). Such receiver devices also exist for computers and come with software that can

permanently synchronize your computer clock with that time signal.

As far I know GPS also sends time information (either that, or the time can be calculated somehow from the GPS information, I'm not too familiar with the GPS protocol). So attaching a GPS receiver to both computers can probaly also get them synchronized to the millisecond. If your synchronization is done via the Internet however, don't expect a better synchronization than one computer being at most 20 milliseconds off.

To update on the commenter,
NTP is not that accurate as people love to claim here:

NTP can usually maintain time **to within tens of milliseconds** over the public Internet, and can achieve better than one millisecond accuracy in local area networks under ideal conditions.

Source: [Wikipedia](#)

I would rather keep them all in sync without any network involved and farther keeping them in sync to the official GMT time and here GPS is probably the only way to get really accurate results on all machines (and that not only down to the ms, actually down to microseconds).

Share Improve this answer edited Jun 3, 2014 at 17:57

Follow

answered Sep 18, 2008 at 23:17



Mecki

133k ● 34 ● 259 ● 269

10 While this answers points out some challenges to a good protocol, the conclusion is completely false. Using NTP you can easily hold machines on a local LAN to sub millisecond precision. Even over the Internet, you can expects to hold time in the say 10 ms range using NTP – [Tall Jeff](#) Sep 18, 2008 at 23:27

11 NTP (the protocol) is capable of and can easily hold better than 1 millisecond precision between machines on the same LAN. I voted you down because what you describe is not accurate. – [Tall Jeff](#) Sep 22, 2008 at 15:18

1 Mecki, you quoted a link that contradicts yourself, because it says "There are several articles on NTP accuracy. The best of them seems to be one by Dave Mills (the primary investigator on NTP). NTP v4 with kernel mods to support it, is capable of much better than 1ms accuracy, possibly as good as 1ns." I think the way to go is the one xsaero00 says, with that authoritative server having a GPS syn card... – [Andor](#) May 5, 2009 at 11:51

4 "According to his article, NTP v3 is accurate to 1-2ms in a LAN and 10s of ms in WAN nets." speaking of the Mills article. Sounds like NOT better than 1ms to me. To get the better quoted results you need "NTP v4 with kernel mods to support it". Unless the linked article didn't summarize the Mills article properly. – [Mark Ransom](#) Oct 7, 2009 at 22:44

1 You seem to be assuming it is WAN use case. But question does not say that -- just "between machines", which I would think is as likely to be within LAN than not. So to me it seems that it is quite possible that 1-2ms is quite within "millisecond" accuracy (I do not read that as request to be accurate within fraction of millisecond) – [StaxMan](#) May 17, 2010 at 19:23



3



I use NTP throughout the whole network at my company and it works rather well. The key is to have one authoritative server on a local network and have every machine on the network synchronize with it. The best is to have a radio clock installed on that server. NTP is great because it does not just correct the clock once in a while, but it actually calculates and correct clock frequency making it more accurate.

Once I had NTP setup on the network I opened like five VNC session to different server and sat there watching the clock. The clocks on all server were in sync withing milliseconds, and this is right after setup. It gets more accurate as it runs.

Share Improve this answer

answered Sep 18, 2008 at 23:13

Follow



Mike Starov

7,266 ● 7 ● 38 ● 37



2



Solutions based on NTP or SNTP can work very well, but it strongly depends on how well the client is implemented.

Certainly, the answer to this question is **not** to use the default Windows time service if you want sub second precision. It is notoriously poor at maintaining a stable time base on a machine and will typically overshoot corrections and is almost unstable even, especially when machines have fairly inaccurate timebases to start with - which is common. Assume the standard built in Window's tools can hold accuracies reliably there to typically only

several seconds between all machines and I typically see swings of as much as 30 seconds between machines - even if you tweak the registry settings.

The freeware tool [Achron](#) is a pretty good solution to get down into the plus/minus 500 millisecond kind of range. Doing better than that will require a more industrial strength solution such as something from [Greyware](#)

Share Improve this answer

answered Sep 18, 2008 at 23:15

Follow



Tall Jeff

9,984 ● 7 ● 46 ● 61

Relative to a good source of time, the NTP pool is a good place to start. See www.pool.ntp.org – [Tall Jeff](#) Sep 18, 2008 at 23:19

NTP is not accurate to the millisecond, you are not following the questioners constraints. – [Mecki](#) Sep 22, 2008 at 9:44

NTP (the protocol) is perfectly capable of holding accuracies between machines on a LAN to better than 1 millisecond. – [Tall Jeff](#) Sep 27, 2008 at 13:21

2 I used: meinberg.de/english/sw/ntp.htm when looking for a good NTP solution under windows – [Sembiance](#) Jun 26, 2009 at 14:32



I've researched (read Googled) on this topic lately and here is what I have learn so far:

2



- To get millisecond-accuracy (or better) you need hardware support. GPS source or hardware time-stamping (and a good time source) in PTP.
- Hardware time-stamping in PTP is done with supported NIC - Intel has them.
- Without hardware time-stamping the accuracy between NTP and PTP are similar.
- (Not used PTP before) I read that NTP is easier to setup.
- My limited experience with GPS time source (over serial) varies. It works great if you can get it to work but there is a device that we have in a data center that I never managed to get it to work...
- If your machines are in colo ask your DC what they can provide - so you don't have to decide.

Share Improve this answer

Follow

edited Nov 1, 2023 at 2:06



Asclepius

63k ● 19 ● 186 ● 156

answered Feb 26, 2016 at 1:27



Lester Cheung

2,030 ● 18 ● 17



1

NTP is definitely the way to go. Basically fire-and-forget, as long as you let it out the firewall on your local master (which is typically the firewall or router machine.)



Share Improve this answer

answered Sep 18, 2008 at 23:08

Follow



Chris

4,956 ● 1 ● 24 ● 17



1

As you've already suggested, NTP is the industry standard solution to this problem, but it either requires Internet connectivity or a stratum 0 source (an accurate hardware clock, like a GPS receiver with a computer interface).



If you're using Internet connectivity, consider using the [NTP Pool](#).



Share Improve this answer

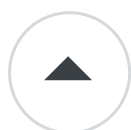
answered Sep 18, 2008 at 23:08

Follow



nsayer

17k ● 4 ● 36 ● 52



1

Keep in mind as well, that the hardware system clock (i.e. the inaccurate one) is only read when the machine starts up - if you're talking about server machines, you're not going to lose time because of them.



Share Improve this answer

answered Sep 19, 2008 at 0:58

Follow





Ana Betts

74.6k ● 16 ● 143 ● 210

Are you sure about this? Could you add some proofs or explanations, please? Thank you. – [Cristian Ciupitu](#) Sep 19, 2008 at 1:04

- 2 I don't have a specific reference, but I do believe the statement is true about reading once. Although, the software real-time clock is probably even a less precise timebase than the hardware one on many machines. Hence the need for an external (say NTP) source. – [Tall Jeff](#) Sep 22, 2008 at 15:17
-