Clock

This page is an attempt to collect all the information about clock and timing updated at December 2011.

You can read the full thread on the freeswitch-dev mailing list with the subject: "Epoch current time difference" http://comments.gmane.org/gmane.comp.telephony.freeswitch.devel/4304

Anthony Minessale says:

FS is keeping it's own time using the monotonic clock in the kernel which is the same mechanism as the system time so your machine probably keeps bad time.

Allowing any software to trust the system time is very dangerous. Something that was depending on the system time to calculate a duration etc can be considerably compromised if the time suddenly changes by hours or years in either direction. Things like daylights savings time can break CDR etc.

The clock in FS is the onboard monotonic clock and the offset to real time is preserved. If you want FS to sync to system time when desired there are commands to perform this operation.

Michael Giagnocavo says:

Daylight saving should never modify the time, as for anything like this, one should be using UTC time, or local time with a UTC offset. Any software that computes a duration differently because of DST cannot blame the system: it is very, very, broken.

The only reason I can think of for the time values to jump around is if your clock is way out of sync and you force a large NTP jump. But in that case, you’re sorta out of luck anyways.

So what if you use NTP server on your system? AFAIK on virtual environments such as vmware even if you install vmware tools you still have to sync to a NTP server to keep your time correct. For more info on timekeeping try for example a google search like: "vmware timekeeping".

Anthony Minessale says:

Most likely you are using something like ntp to sync your system clock, if so you should also sync the FS clock (e.g. fsctl sync_clock) when you change the ntp time. FreeSWITCH will only run as good as a VE can perform.

It is perfectly safe to use it in production but we WILL NOT support performance issues related to running in a VE because there are too many variables and emulations present to know the root cause of any particular performance issue. Our community conference server is a VE but its openvz which has proven the most reliable VE to run FreeSWITCH under that we have tested.

Especially with current FS HEAD, it runs as expected on EC2 assuming you are choosing the correct image with full CPU power and the amazon linux OS.

There are some startup options and commands which are involved with this topic. Startup options explained by Anthony:

- `nocal` turns off the calibrating clock sequence executed at startup. It does several tests to find the number of microseconds you should sleep to get exactly 1 millisecond. Some systems are off in this respect and the test finds the optimal value, when you specify `-nocal` it just uses the default of exactly 1000 microseconds.

- `nort` essentially disables clock_nanosleep from being used as a method to sleep small amounts of time. On many systems this function is by far the most accurate but on many others its not. Its harder to test for this so we have to figure it out manually, really its mostly older boxes who need this.

On windows the default behaviour is a little different as explained by Peter: About Windows - yes, it will work as on Unix systems if starting with `-monotonic-clock` - it will use a true monotonic clock.

The commands to use to sync the FS clock with the system clock should be `fsctl sync_clock (mod_commands#sync_clock)`

There is another command to do the same "gracefully" when there are no active calls: `fsctl sync_clock_when_idle`

You can use this command to check FreeSWITCH's date and time: `strftime (mod_dptools#strftime)`. It returns the local system time. If format_string is not specified, it defaults to "%Y-%m-%d %T", eg. 2010-04-28 17:47:29