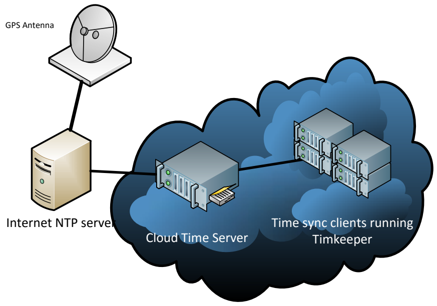
Leap Second coming December 2016, already causing problems

Leap seconds have caused tremendous problems with IT systems in the past. The last few, in 2015, 2012 and 2008, resulted in thousands of servers crashing across the internet leaving several sites (Foursquare, Yelp, LinkedIn, Gawker, Slashdot, Mozilla and Quantas Airlines) out of service for a while. Applications using Cassandra or Hadoop also saw major service outages requiring rolling restarts due to mishandled leap seconds. The bugs in free time synchronization software and the Linux kernel that caused this are still out there, still installed on many systems, waiting to be triggered yet again on December 31, 2016. FSMLabs TimeKeeper® suite includes hardware and software that supports PTP, NTP and many other protocols for time synchronization. We handle leap seconds correctly and are able to avoid the Linux kernel bugs that have caused system crashes. In this whitepaper we detail how we properly handle leap seconds predictably, how we avoid Linux bugs and how we can keep your systems working properly even with invalid leap seconds and similar malfunctions of other products on the time network.

Time Server Setup

Typical setup for a datacenter getting time from a GPS based appliance.



What a Leap Second is

A leap is similar to a leap year. Instead of adding one day to the end of a month periodically, a leap second is a single second inserted in the last minute at the end of a day. The next one is December 31, 2016 at midnight UTC (Saturday evening for the USA, Sunday late morning for Australia and Asia). That means the last minute of the day has 61 seconds instead of 60. It is possible to remove a second instead of adding one as part of a leap second but that has never been required. This extra second in the day is meant to align standard time with the movement of the earth and sun.

How to handle leap seconds

After a leap second occurs computer clocks will be one second ahead of the correct time. The two most common methods for handling this are 'stepping' and 'slewing'.

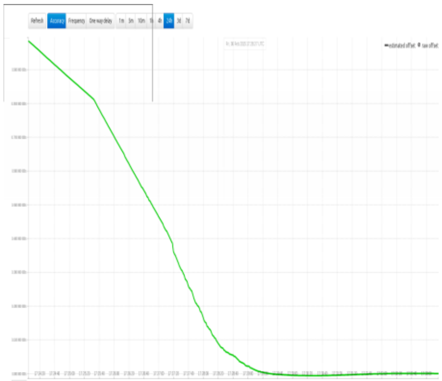
A 'slew' is where a clock is slowed down until the clock catches up with the correct time and then restores the clock speed to normal. This has the advantage of not introducing a jump backward in time which can affect software and operating systems (which can trigger Linux kernel crashes). Depending how it's implemented and how much care is taken to keep all systems aligned it's possible that this slew can take a very long time during which time is incorrect.

It is also possible for multiple systems on a network to slew at different rates which means during this slew time across the network is not only incorrect but networked systems are also out of sync with one another.

Google got some publicity in 2012 by claiming they had discovered the 'slew' method, redefined it as 'smearing', but in fact it has been in wide-spread use since the first leap second in 1972. The Google innovation was that they take 24 hours to correct out the leap second which is only possible because accurate time is less important to their application than is all of their systems agreeing on time. Most applications are far more demanding of both correct time and agreement on time across the network.

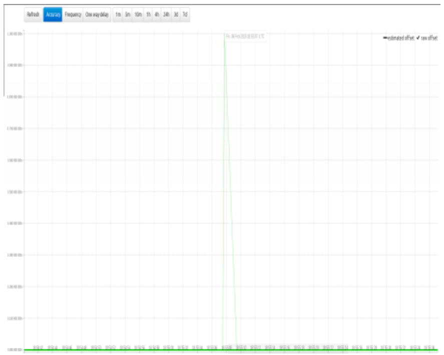
TimeKeeper Slewing

Correcting a one second time offset gradually



TimeKeeper doing a Step

Correcting a one second time offset immediately



A 'step' is where the clock is reset to remove the one second error instantly. The downside of this method is that the system clock goes backward by one second. This has caused Linux kernel crashes in the past. The 'step' correction method can also cause some programs to misbehave since they do not expect time to go backward. FSMLabs strongly recommends customers rely on the default slewing technology where possible in TimeKeeper in order to protect applications from the unknown effects of a backwards jump in time.

TimeKeeper is able to use either method. Even when doing a 'step' TimeKeeper avoids triggering Linux kernel bugs that cause crashes unlike some open-source alternatives. Typically customers prefer the 'slew' since it only lasts about 5 minutes, we ensure that clients on a network are slewing together so that they all agree on time during that period and it avoids time jumping backward by one second.

2016 Leap Second is already causing problems

Even though the 2016 leap second has not yet happened it has already started causing problems. In late July 2016 the GPS constellation correctly started advertising that a leap second would occur this year. Many GPS receivers and time serving devices interpreted that incorrectly and assumed that a leap second would occur immediately instead of 6 months later. In addition to buggy hardware and software prematurely implementing a leap second a few pay-for-subscription time services began to offer bad time. At the time of this writing a number of appliances in production networks are incorrect by one second. Some of those appliances are public NTP servers on the internet and are providing incorrect time. Even worse, open-source software that was designed without defense against incorrect time updates is propagating incorrect time even further.

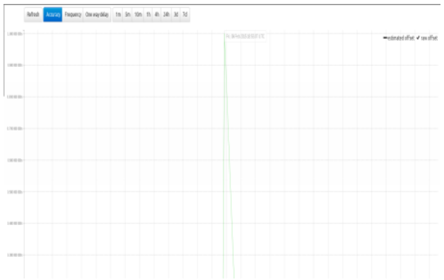
Without a time infrastructure built for defense many sites are vulnerable to a single erroneous time source.

Defense against bad leap seconds

Most legacy time synchronization software will accept a leap second when told to insert one without checking the validity of that command at all. This vulnerability is currently causing bad time to propagate throughout the internet which is exposing systems to leap second related issues long before the leap second was meant to occur.

TimeKeeper rejecting false leap second

Rejecting primary time source with a bad leap second and using secondary sources



The open-source community has seen the dangerous state of the current situation for over a year and there does not appear to be a consensus approach. As a result there are numerous rewrites, new projects and proposed fixes in progress.

One method we use to defend against bad time is the "sourcecheck" feature of TimeKeeper. In addition to detecting GPS spoof attacks, bad oscillators and other kinds of common time errors it detects false leap seconds and stops them in their tracks. TimeKeeper can compare multiple time sources to validate time and frequency over many different protocols and technologies in order to detect bad time. That includes leap seconds that are introduced at the wrong time or missing when they should happen. If a primary time source is found to be in error TimeKeeper will switch to the next highest priority time source and send alerts about the event (via SNMP, email, web alerts, syslog). These alerts allow IT staff to look into the issue and deal with the underlying problem since TimeKeeper automatically switch to back-up time sources. TimeKeeper also logs all events and timestamps for auditing and analysis later.

TimeKeeper has been managing time reliably for years

TimeKeeper has been working reliably for years in everything from financial exchanges, tier one banks all the way to military systems. With and without leap seconds it has kept time consistently for customers. It does that because security and protection are built into the software from the ground up. This is done with a combination of tools and techniques that starts with extensive automated testing which includes leap seconds and other extreme situations to make sure behavior is predictable, quick and correct.

Avoid 2015, 2012 and 2008 repeat

To avoid a repeat of the previous failures and service outages we recommend extensive testing before the leap second happens. This won't protect you from false (early) leap seconds introduced by faulty software but testing will give you some idea of what will happen this June. Failing to test may leave you vulnerable to the same problems from years past (just google "2012 leap second web"). Many fixes have been applied to the problematic software and applications but if you haven't updated, applied patches or confirmed that you have the updates with your vendor it may be time to start doing that.

TimeKeeper provides the capability to simulate leap seconds so that you can see what will happen to your infrastructure. If you discover troublesome applications that cannot handle an abrupt change in time from a leap second without extensive re-working or patching then TimeKeeper is able to slew the time to correct the leap second offset without disrupting operations.

How to Purchase

TimeKeeper, TimeKeeper Server Software, and TimeKeeper Client Software are all available from FSMLabs and FSMLabs' resellers. For purchase information or for a live demonstration of TimeKeeper please contact FSMLabs at sales@fsmlabs.com.

TimeKeeper and FSMLabs are registered trademarks of Finite State Machine Labs