



Requirements for UTC and  
Civil Timekeeping on Earth:  
A Colloquium Addressing a  
Continuous Time Standard,  
May 29-31, 2013

# Time, Timestamps, Timescales

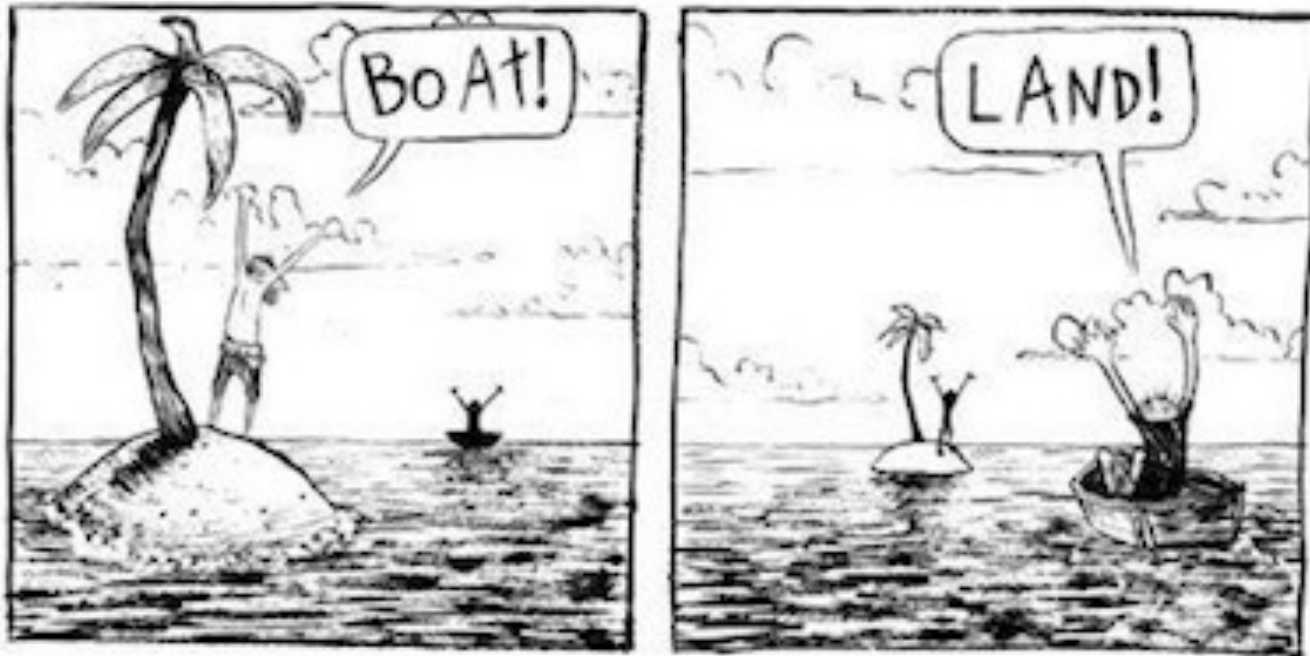
Some altitude, perspectives and  
ideas



# Who/What is NTF?

- The NTP Project needed the support and backing of a legal entity.
- I felt an “NTP Foundation” would have insufficient scope.
- Network Time Foundation seemed “right”.
- NTF currently supports NTP, PTPd, GPSPD, LinuxPTP, and RADclock.
- NTF is applying for 501(c)3 status.

# Time



Perspective...



# Perspective...

- I'm a fan of “robust mechanism” and keeping “mechanism” separate from “policy”.
- I try and remember the Pig and Chicken analogy that the Agile folks use.
- IMO, don't break in-use stuff. If the in-use thing sucks for some folks, offer a replacement and remember my favorite between-the-lines Bible quote: “Blessed are those who get what they deserve.” This cuts several ways.



# Time

- “Knowing the time” is important because we can note when something happened and we can plan when to do something.
- This requires dissemination of the time and timestamps that contain “enough” information to be generally useful.
- “Time” means different things to different people.



# Time

- It was easier to use “time” when the wobbles were in the noise.
- Timescales solve problems for different groups.
- Problems occur when different groups disagree about “signal” and “noise”.
- Larry McVoy says: In theory, theory and practice are the same. But in practice they are not.



# Timestamps

Current timestamps are mostly OK for “local use”. Mostly.

- seconds since some epoch
- <days since epoch>, <seconds since midnight>
- YYYYMMDD-HHMMSS - Long-standing hospital database does not bill millions of dollars each Fall’s daylight-savings correction



# Timestamps

- Monotonic time and databases
- System time may be known to be undergoing a correction.
- Error bounds?
- What timescale is being used?





# Timestamps

- Will a “boot counter” be useful when comparing timestamps across a reboot?
- Will a “host ID” be useful when comparing timestamps between multiple systems?
- The NTP 32/32 timestamp format is good for 136 years’ time with a resolution to 233 picoseconds. A 40/32 timestamp is cumbersome but probably sufficient. 64/32 or 64/64 timestamps are convenient and large.



# Timescales

- There are a lot of them.
- Some change never, or rarely (GPS)
- Some change often (IERS-a)
- Local timezones are a sub-case
  
- Google's "leap smear"
- Martian Standard Time



# Timestamp Structure

- System time
- Amount of pending correction
- Expected/Maximum error
- Timescale
- Boot counter/nonce?
- Host ID?
- Version number



# Timestamp Example

{System Time, Offset to Correct Time}

UTC	NTP-DLM	NTP-Windows	Smear24H
00:00:00.00	{00:00:00.00, 0}	{00:00:00.00, 0}	{00:00:00.00, 0}
06:00:00.00	{06:00:00.00, 0}	{06:00:00.00, 0}	{05:59:59.75, .25}
12:00:00.00	{12:00:00.00, 0}	{12:00:00.00, 0}	{11:59:59.50, .50}
18:00:00.00	{18:00:00.00, 0}	{18:00:00.00, 0}	{17:59:59.25, .75}
23:59:59.00	{23:59:59.00, 0}	{23:59:59.00, 0}	{23:59:58.00, 1}
23:59:59.50	{23:59:59.50, 0}	{23:59:59.50, 0}	{23:59:58.50, .5}
23:59:60.00	{23:59:59.99, .0}	{23:59:59.99, .0}	{23:59:59.00, .0}
23:59:60.50	{23:59:59.99, .5}	{23:59:59.99, .25}	{23:59:59.50, .0}
23:59:60.99	{23:59:59.99, 1}	{23:59:59.99, .50}	{23:59:59.99, .0}
00:00:00.00	{00:00:00.00, 0}	{00:00:00.00, -.50}	{00:00:00.00, 0}
00:00:00.50	{00:00:00.50, 0}	{00:00:00.50, -.25}	{00:00:00.50, 0}
00:00:01.00	{00:00:01.00, 0}	{00:00:01.00, 0}	{00:00:01.00, 0}



# Putting it to Use

A new timestamp structure is only useful if it can be widely and generally portable:

- Kernel support
- Library support



# Adjusting System Time

- Forward:
  - Adjust “system time”, or
  - Increase “amount of pending correction”
- Backward:
  - Decrease “amount of pending correction”

The OS applies pending corrections according to its policy rules.



# Adjusting System Time

- Forward time adjustment seem to be pretty straightforward.
- Backward time adjustments are more challenging, as monotonic time is generally “good”.

If we want to step the time backwards, make tiny advances to the system clock when needed and decrease the pending correction by 1 second per second.



# Using timestamps

The timestamp library API needs to handle:

- Adding/subtracting timestamps
- Canonicalization of timestamps
- Comparing timestamps
- Converting timestamps (timescales)





# Timescale Database

I'm operating on the belief that a timescale database won't be that much harder to implement and maintain than Arthur David Olson's Timezone Database.



# Making it Go

I wasn't planning to start working on the "improved timestamp" project so soon, because \*I\* wanted to work on it and I'm being Busy. And then GSoC happened, and a bright student was looking for a new project to replace the dud project I originally proposed...



# Making it Go

So by the end of the summer we should know a lot more about:

- Pros/cons of the new timestamp format
- Kernel implementation issues
- Issues around timescale specification
- Performance of the timestamp library



# Ongoing Questions

POSIX mostly decided upon absolute event timers. It would be nice if we could come up with a mechanism to notify a process that the system time had changed so that the application had a way to decide if it wanted to adjust its events or not.



# Help NTF Help You

Please visit <http://nwtime.org> and learn more about Network Time Foundation.

Join NTF and invite others to join, too!

Help NTF help you!