Rise of TrueTime: Rationale behind Amazon Time Sync Service

Abhishek Tiwari

Citation: *A. Tiwari*, ”Rise of TrueTime: Rationale behind Amazon Time Sync Service”, Abhishek Tiwari, 2017. doi:10.59350/mpztp-jvn26

Published on: December 01, 2017

At re:Invent 2017, Amazon Web Services (AWS) announced Amazon Time Sync Service which provides a highly accurate and reliable global time reference. This service is immediately available in all AWS regions.

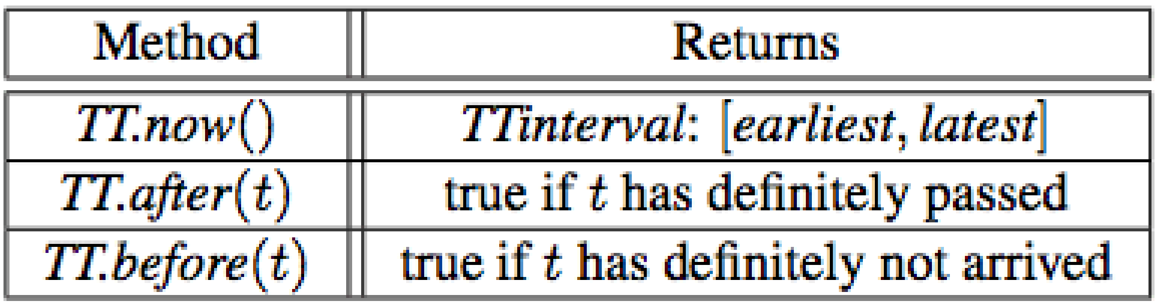
Amazon Time Sync Service

In many ways, Amazon Time Sync Service is truly inspired by Google' TrueTime. TrueTime is a global reference clock with a bounded non-zero error. TrueTime utilizes satellite-connected GPS and atomic clocks. Google's data centers are equipped with a series of GPS receivers and atomic clocks. With the launch of Amazon Time Sync Service, AWS regions are equipped with similar GPS and atomic clocks.

So what really inspired Amazon to replicate Google' TrueTime? Before we answer that let's have a quick look why Google's TrueTime was introduced at first place. TrueTime was introduced to support Spanner - Google's globally distributed, and synchronously-replicated database. Under the hood, TrueTime uses two forms of time references - GPS and atomic clocks - each with different failure modes.

Spanner database

Although a Spanner database runs across multiple servers located across multiple datacenters, TrueTime makes it semantically indistinguishable from a single-server database. TrueTime to generate monotonically increasing timestamps which Spanner uses write transactions and strong reads. TrueTime provides various API calls and `TT.now()` method returns a time interval. When two intervals do not overlap, then we know calls were definitely ordered in real time [1].



Method	Returns
$TT.now()$	$TTinterval$: $[earliest, latest]$
$TT.after(t)$	true if t has definitely passed
$TT.before(t)$	true if t has definitely not arrived

Figure 1: TrueTime API. TrueTime explicitly represents time as a TTinterval

In Spanner, TrueTime is used to guarantee strictest concurrency-control for transactions at the global

[1] Spanner: Google's Globally Distributed Database

scale - a concept Google refers to external consistency. It is important to note, that external consistency is a much stronger property than strong consistency which in turn is stronger than eventual consistency. Spanner achieves external consistency, consistent reads without locking, and consistent snapshots by utilizing the TrueTime[2].

Along with Amazon Time Sync Service, AWS also announced the new database-related enhancements - Amazon DynamoDB Global Tables and Amazon Aurora Multi-Master. I for one strongly think that under the hood these enhancements are utilizing Amazon Time Sync Service but without external consistency. That said, DynamoDB Global Tables don't offer external consistency - in fact, it barely does eventual consistency. In case of strongly consistent reads, it requires the same region for reading and writes [3]. Several people have argued that replicating TrueTime requires more than simple global synchronization which is what so for AWS has achieved [4].

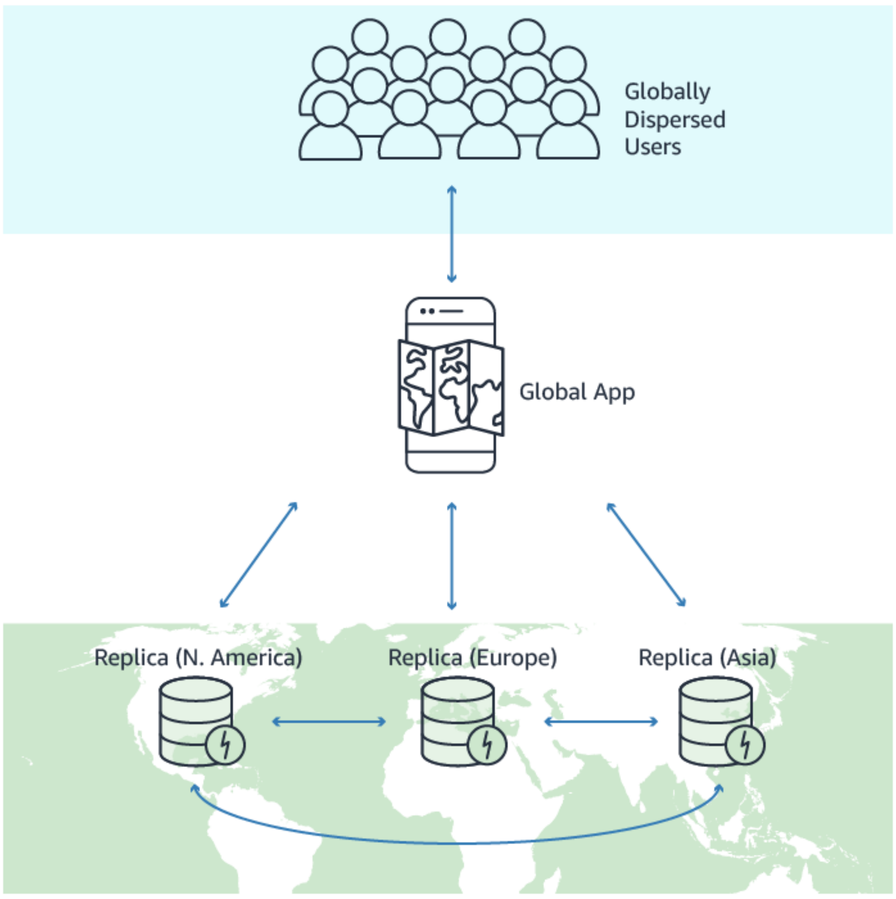


Figure 2: Amazon DynamoDB Global Tables: multi-region, multi-master tables

[2]Spanner, TrueTime & The CAP Theorem
[3]Global Tables: How It Works
[4]Spanner, the Google Database That Mastered Time, Is Now Open to Everyone

Obviously, there are a large number of distributed computing problems which require accurate time reference and databases make a small subset of these problems. Having a global time reference is easy but how you use them to ensure higher data consistency and to resolve conflicts in a global master-master database setup is a hard problem.

Although AWS has not shared many details about the implementation of Amazon Time Sync Service. I think under the hood AWS has replicated the core ideas of TrueTime infrastructure[5] with some variations like smoothing out leap second.

> TrueTime is implemented by a set of time master machines per datacenter and a time slave daemon per machine. The majority of masters have GPS receivers with dedicated antennas; these masters are separated physically to reduce the effects of antenna failures, radio interference, and spoofing. The remaining masters (which we refer to as Armageddon masters) are equipped with atomic clocks. An atomic clock is not that expensive: the cost of an Armageddon master is of the same order as that of a GPS master. All masters' time references are regularly compared against each other. Each master also cross-checks the rate at which its reference advances time against its own local clock, and evicts itself if there is substantial divergence. Between synchronizations, Armageddon masters advertise a slowly increasing time uncertainty that is derived from conservatively applied worst-case clock drift. GPS masters advertise uncertainty that is typically close to zero.

Closing thoughts

Amazon Time Sync Service is available at no extra cost for customers using EC2 instances in Amazon Virtual Private Cloud (VPC). The service is available through NTP via a universally reachable IP address `169.254.169.123`. To use this service, AWS recommends to uninstall your default Network Time Protocol (NTP) package first and then install Chrony which is an alternative implementation of the NTP. After this, Chrony will synchronize the EC2 instance's system clock with universal reference clock endpoint `169.254.169.123`.

[5] Spanner: Google's Globally-Distributed Database