FSMLabs
Enterprise Real-Time

CLOCK SYNCHRONIZATION IN FINANCE AND BEYOND

VICTOR YODAIKEN

1. INTRODUCTION

FSMLabs produces a suite of clock synchronization products under the name TimeKeeper®, primarily for customers in financial trading but also for a wide range of customers in other fields: from first responders to internet gaming providers, managers of big data systems, and radio/TV recording and broadcast. Clock synchronization is an emerging required tool for any area of high speed transaction processing or coordinated computing and data aggregation (see section 3). TimeKeeper is widely acknowledged to be the market leading technology in finance, which has stringent accuracy and availability requirements.

- **TimeKeeper Client** software runs on an application server computer or virtual machine (Linux, Windows, Solaris) and locks the system clock to a reference clock time that is sent over packet networks.
- **TimeKeeper Server** software has TK Client as a base and adds the capability of serving time to clients and monitoring their accuracy and other properties. TK Server can get time over a network or from GNSS receivers for satellite systems such as GPS or Galileo.
- **TimeKeeper Compliance** software has TK Server as a base and archives and provides search and reporting capability for time accuracy information from clients and sources, primarily to assist financial services companies address regulatory requirements, but more generally to assist central IT teams manage large scale clock synchronization networks over long durations.
- **TimeKeeper GrandMaster** is a high end server computer based appliance with TK Server, a GPS/Galileo/GNSS receiver and high quality oscillator that allows it to keep time steady even during interruptions of satellite time service.

TimeKeeper uses and gets high accuracy and interoperability from both of the standard network clock distribution protocols: the Network Time Protocol (NTP) and IEEE 1588 Precision Time Protocol (PTP). Innovations include: machine learning algorithms to compensate for the noisy operation of commercial networks, fault tolerance that operates above the protocol level, data visualization and analysis, real-time instrumentation and alerting, whole network data gathering and consolidation, and sophisticated network operation to make communications more deterministic. As an example of data visualization, the TimeKeeper "SkyMap" produces a visualization of radio signal reception (figure 7) that can be used to identify failures and detect security problems with GPS and other satellite clock sources.

Date: Nov 21-June 6 2017.
Key words and phrases. programming, sequences, sorting.

TimeKeeper adjusts the operating system clock to match the reference clock so that application programs will see accurate time with no changes to application code. Production systems where client clocks synchronize to the reference clock within 200 nanoseconds are common in well provisioned modern data centers even for virtual machine instances. In more challenging circumstances, a few microseconds of accuracy is more usual. TimeKeeper is designed to squeeze as much accuracy as possible from any environment and it uses patented technology to use monitor multiple clocks in real-time to improve accuracy and to offer fault tolerance. The underlying technology is explained in more detail in section 4.

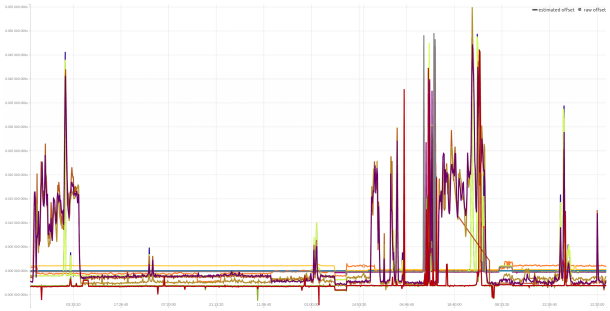

FIGURE 1. Multiple sources

2. CLOCK SYNCHRONIZATION IN FINANCIAL TRADING.

2.1. **Timestamps.** Before the era of electronic trading, financial trading firms made extensive use of mechanical devices that stamped a piece of paper with the current time. These devices allowed financial trading firms to learn patterns in the market, check efficiency of trading partners, exchanges and brokers, and protect against malfeasance. As an example: a rogue trader who wanted to profit by anticipating changes from pending customer orders ("frontrunning") would be exposed by timestamps showing his or her orders appearing in-between the arrival of the customer order and its submission to the exchange. Timestamps on electronic orders serve exactly the same purposes but because of the high speed of digital transactions and the distributed nature of electronic trading, clocks have to be exceptionally precise to make timestamps useful.

FIGURE 2. TimeKeeper GPS clock with 40 nansoseconds precision

If a server computer sends an order and a second server computer belonging to the same firm receives the confirmation and the clock on the second computer is 100 microseconds behind the clock on the first computer, the timestamps might show the

confirmation arrived before the order was sent. This would be hard to explain to a customer or regulator. Now suppose there are thousands of computers trading in Singapore, Chicago, London, and NY for that one firm and consider what the books are like if the clocks are wandering. Out of sync clocks are particularly deadly for automated trading and analysis algorithms that look for correlations between events on the market - fuzzy timestamps obscure patterns or create imaginary patterns. And for regulators trying to make sense of the interactions of many thousands of market participants, fuzzy timestamps make the job impossible. Precisely synchronized clocks are the foundation of data integrity in electronic trading - and as transaction execution becomes faster and electronic trading spreads into additional asset classes, the requirement for clock synchronization becomes only more demanding. The well known book on high frequency trading, Flash Boys[15], explains how firms with poor quality clocks can get completely blindsided by competition and market changes.

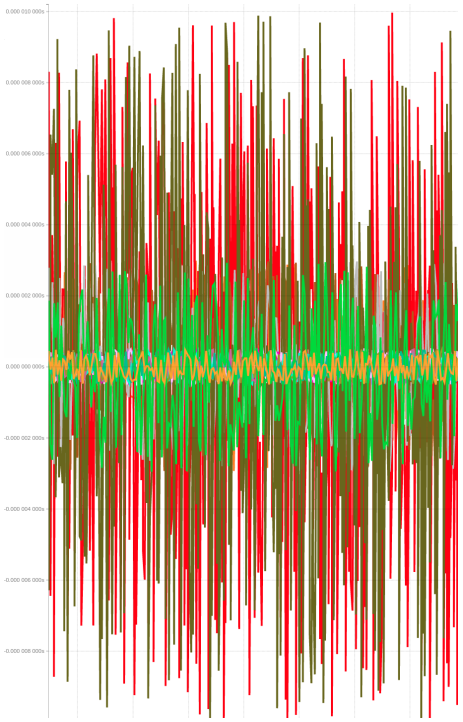

FIGURE 3. TimeKeeper Server monitoring clients that are within 8 microseconds of reference time

2.2. **Regulations.** To force the books to be consistent, the new MiFID2 regulations in the European Union and UK[7] require many market participants to make sure all clocks involved in high speed trading (not necessarily just high frequency traders) synchronize all their clocks to the official UTC time within 100 microseconds. FINRA/SEC have recently tightened US rules to require clocks be synchronized to within 50 milliseconds of US official time (NIST) as well. Further tightening seems inevitable. Firms with poor timestamp accuracy are not only asking for regulatory problems, but are not capable

of evaluating their own operations or detecting fraud or even of enforcing SLAs with network providers and trading partners.

2.3. **Record Keeping.** Although the changes in required precision in the new regulations got the most notice, requirements for documentation and for making sure that clocks are always precise were more significant. Even the old FINRA "one second" standard was routinely violated in the past using single source, poorly monitored clock sync methods with weak instrumentation. To address these issues, market participants need

FIGURE 4. One of the TK-Compliance reports

an integrated approach to monitoring and record keeping that extends from clients that can monitor multiple clock sources and share that data with servers, to servers that monitor their own sources as well as their clients, to some compliance system that puts all that data into a searchable database, generates reports, and produces archives for long-term storage.

FIGURE 5. TimeKeeper tracking 2 clock sources within 90 nanoseconds

2.4. **Fault tolerance and instrumentation.** One issue for regulatory compliance is that clock synchronization software is notorious for under-reporting errors and silent failures. For example, NTPd[5], a NTP client often bundled with standard operating systems, may continue to report that the clock is near perfect in a virtual machine that has just been suspended for several seconds where it is clear that the clock is now several seconds behind the actual time. The free software PTP client, PTPd, will often fail to detect network configuration problems that make high quality synchronization impossible but report good sync nevertheless. In fact, the whole EUREX trading network was inoperable for several hours in 2013 due to a minor failure in a single satellite receiver clock.

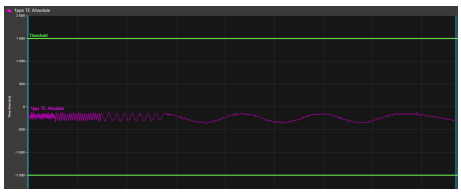

FIGURE 6. TimeKeeper accuracy test using Calnex Paragon PTP test device

TimeKeeper provides a pessimistic estimate of accuracy and looks for error conditions so it can generate alerts, but even TimeKeeper can't detect all errors. In particular if all there is a single clock source that is wrong but has stable frequency, no client software will be able to detect the error (which is why FSMLabs recommends multiple independent clock sources for critical systems). Figure 1 shows 7 days of accuracy measurements on multiple sources - the big swings on some of the sources illustrate why multiple sources are so necessary.

Many FSMLabs customers set up their systems so that if, a clock at one data center in the New York area loses its connection to GPS or loses confidence in the signal it is getting from GPS, the clock can fall back on time from a second clock in another data center. A system where multiple sources, perhaps using several physically separate antennas and an additional terrestrial source or cross-link is going to be much more reliable than one with a single source. Internal clock servers and clients should also be set up with redundant clocks.

2.5. **Traceability and Satellite time.** Regulators generally require that clocks be "traceable" back to some official source. In the European Union, that source is UTC which comes out of the International Bureau of Weights and Measures in Paris and is a blend of the official times that come from national physics labs like the National Institute of Standards and Technology (NIST) in the USA and the Physikalisch-Technische Bundesanstalt (PTB) in Germany. The European regulator has declared that satellite time is a satisfactory substitute for UTC (which is ultimately determined weeks after the fact in any event). The US regulators at FINRA and the SEC usually cite NIST as the offical time, but consider GPS time to be good substitute (in fact, they prefer it to the low quality of NIST time available over the Internet.) In practice, satellite clocks and national lab clocks and UTC are all within a few nanoseconds of each other, and the national labs produce official validation of that accuracy to make satellite time "traceable" to UTC/NIST.

3. BEYOND FINANCE.

Telecommunications was one of the earliest fields to incorporate precise time distribution and requirements for synchronized clocks continue to be an important part of system design, especially with faster systems such as 5G cellular telepones. Newer application areas include internet gaming, manufacturing, and radio and TV broadcast and electric power distribution[13]. One of the most interesting areas is in making big distributed software systems work fast reliably.

The role of accurate timestamps in distributed consensus has been known at least since the 1970s[19] and it is common for distributed databases to require at least some level of clock synchronization. In a famous academic paper from the 1980s, Barbara

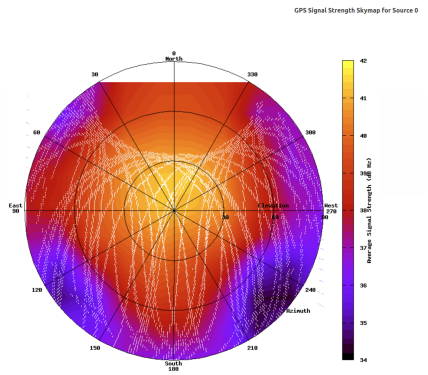

FIGURE 7. SkyMap

Liskov looked at the use of synchronized clocks to control distributed systems[16] and there has been a lot of recent interest from Google on reusing or extending that work[4, 12]. Google's Spanner database[6] relies on clock sync to reduce data synchronization and there are many other places where synchronized clocks allow coordination without discussion. The basic idea is to replace or reduce reliance on *data synchronization* involving mutual exclusion and locks by relying on clock synchronization. In fact, much of this work adapts or builds on what has already been constructed in financial trading. Essentially, financial trading systems are high speed, distributed, transaction processing systems where clock synchronization is used to assure data integrity in the absence of data synchronization. The need for such systems is quite broad.

4. TECHNOLOGY.

FIGURE 8. TimeKeeper GrandMaster showing 25Gbps network ports and GNSS antenna connector.

4.1. **Clock distribution.** The focus in this note is precise time for software applications. For these applications, it is important to look at the entire clock delivery system from reference source to software application - the end-to-end operation of clock distribution. High quality clocks at some demarcation point do not assure anything about clock quality at point of use. For higher accuracy systems, the normal configuration involves:

- Some authoritative clock or clocks, usually ones that get time a Global Navigation Satellite System (GNSS) such as the US Global Positioning System (GPS) and/or European Galileo. These are called "primary" clocks.
- Distribution of time over packet networks from the primary clocks to client computers and other devices and to secondary clocks which act as intermediaries for larger networks (or to bridge clock delivery).
- Synchronization of client operating system clocks to time distributed over the network so that application programs can access accurate time using the standard APIs.

There are two standard network protocols for distributing time over a network: the Network Time Protocol (NTP [17]) and IEEE 1588 Precision Time Protocol (PTP[14]). Although it is often claimed in marketing and technical literature that PTP is more accurate than NTP, TimeKeeper's implementation can produce high accuracy from both. The underlying mechanisms of the two protocols are similar in many respects, but there are significant differences in manageability. In particular, PTP has a number of weaknesses in large scale networks[11, 18]. One of the reasons for TimeKeeper's dual protocol design is to allow seamless integration into existing networks without expensive network equipments upgrades. On the other hand, the two protocols are well defined enough so that good implementations are interoperable. One of the reasons for TimeKeeper's dual protocol design is to allow seamless integration into existing networks without expensive network equipments upgrades.

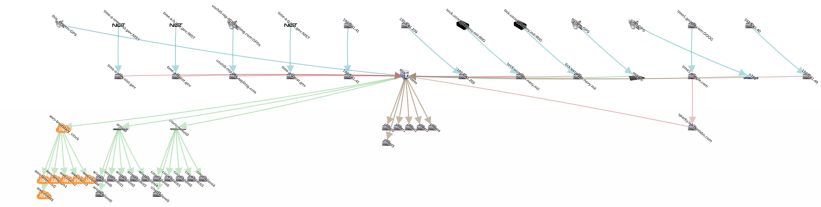

FIGURE 9. TimeMap shows how clock information is distributed

Oscillators on most commodity server computers are poor quality and can drift a microsecond a minute or possibly more so the clients must adjust time at least every few seconds[1]. For both PTP and NTP, the core operation is for a client to request time from a primary or secondary source and then receive an update from that source. The updates from the authoritative clock take some time to cross the network to the application computers and the clients take some time to process the packet and start updating - so the time from the authoritative clock is stale and the clients have to take that into account. One of the biggest challenges for commercial network clock synchronization clients is coping with the noisy signal from the clock source. This variability requires high quality filtering and smoothing to work. Simple implementations, such as those found on many network switches that offer to act as intermediate clocks will sometimes be error amplifiers because they cannot filter effectively. Telecom clock synchronization usually depends on extremely stable dedicated clock distribution connections and for that reason requires radically different algorithms.

[1]However, it is *not* correct that simply increasing the adjustment rate will improve accuracy or that it is always helpful.

PTP primary clocks are called "GrandMasters". NTP primary clocks are called "Stratum 0 Servers". Secondary clocks which act as intermediaries between primary clocks and clients are used in both protocols. PTP secondary clocks are called Boundary Clocks and NTP secondary clocks are Stratum $n+1$ servers with n being the level of the stratum servers [2].

4.2. **Where does time come from?** Clock time is essentially something created by the National Physics labs, using sophisticated atomic clocks. These clocks are aggregated and tuned so that there are national consensus times and then the national clocks are brought into consensus under UTC. The clocks provide both a calendar time (usually in a character string) and a frequency (usually as a pulse per second). This time can then be uploaded to a Global Navigation Satellite System (GNSS) such as GPS and Galileo which incorporate both atomic and maser clocks to stay in sync with the national clocks. Satellite receiver clocks can then receive the time from GNSS systems and synthesize their own clocks[1]. Higher end satellite receiver clocks can easily keep within 50 nanoseconds of satellite time when the signal is strong enough.

4.3. **Holdover and Security of GNSS.** Satellite receiver clocks that equipped with with high end oscillators can maintain high quality time even during interruptions of the signal. The TimeKeeper GrandMaster comes standard with a double oven cooled oscillator that can keep frequencies stable to within 4 microseconds a day, with no adjustments and has is an optional atomic clock that drifts no more than 0.6 microseconds a day. Network clocks can be designed to detect problems with satellite signals and to enter a "holdover" state where they rely on their internal oscillators until signal is restored. Methods such as TimeKeeper Skymap (figure 7) can diagnose interference or compromise.

4.4. **Accuracy in practice.** Our main focus is on clock synchronization for software applications where limitations in execution speed and latency of standard computing platforms determine how much clock precision is usable. Although national labs clocks and UTC differ from each other, the differences are not significant for current and foreseeable computer technology. A cache miss in a Xeon class server computer takes on the order of 90 nanoseconds, network variability is in microseconds, and operating systems introduce delays in milliseconds [2]. Although there are claims of pico-second level clock accuracy from specialized technologies, these claims seem to be made without reference to effects of memory delays or even the reliance on spread-spectrum clocking techniques to limit radio frequency interference in commercial computing devices. Specialized hardware, such as FPGA based trading engines or specialized software such as using real-time operating systems can use higher precision clocks, but even for these systems, the single digit nanosecond differences between the primary clock sources are not significant. Higher precision relative clock synchronization (where multiple devices are synchronized to each other) is achievable with existing technology using simple techniques, such as reserved timing networks and tight control on device workloads.

Unfortunately, claims that NTP has fundamentally lower performance than PTP are still often encountered in the technical literature and marketing material, but these are simply incorrect and fly in the face of the fundamental similarities between the protocols. A typical recent example can be found in [12] which starts of with this claim, apparently confusing the protocol with the the NTPd [3] free software implementation

[2]The PTP naming convention is "slaves" and "masters" but that's both an ugly metaphor and one that lends itself to some design errors as seen below in the discussion of Best Master Clock.

(which *is* limited in accuracy). NIST time is currently available over the Internet via NTP at a level of accuracy that is sufficient, with some care, for clocks that need only be generally within a few milliseconds of reference time and that can tolerate excursions at the level of a second on occasion. The limitations of NIST Internet time accuracy are mostly due to the variability of packet delays in the Internet itself. The UK National Physics Lab has contracted to provide more accurate time over terrestrial networks to data centers in and around London using PTP in a service known as NPLTime. There is no basic technical reason for the choice of PTP over NTP in that service.

4.5. **Networked Clock distribution.** Network performance of primary, secondary, and client devices can have dramatic effects on clock distribution. For example, connecting clock sources that have lower speed network connectivity to a high speed network causes asymmetric packet delivery and clock errors. GNSS Clocks that support only 1Gbps or even 100Mbps network ports are still common and these produce multiple microsecond errors in clock accuracy. The PTP standard attempted to work around this problem with an elaborate technology called *transparent clocks* which has proved highly profitable for network equipment vendors. Transparent clocks adjust PTP packets to incorporate network delays but it is primarily useful on slower and older network equipment and provide no useful information when network elements are symmetric (delays are the same in both directions). It is still common in the literature to see reference to "head of queue" issues in network elements which can cause high levels of asymmetry, but these issues were solved in commercial networks decades ago. Similarly, using "lucky packets" to quantify network delays is obsolete in modern networks. On the other hand, *Hardware timestamping* where the network devices on provide clocks to timestamp packets on transmit and receive is valuable. Hardware timestamps allow sufficiently capable clock synchronization software to factor out operating system network stack delays and variability. This feature is widely available for both NTP and PTP packets and is essential for highest quality clock accuracy.

4.6. **Fault Tolerance.** Because clock synchronization must work *all the time*, reliable clock synchronization depends on a solid model of fault tolerance. The built in model for fault tolerance in NTP is to average multiple sources together - which means that any failure or blip on any source causes an error. There is a second mode available in some NTPd versions which avoids averaging, but it is quite fragile and complex - and it defeats "traceability". PTP has a required mechanism called Best Master Clock (BMC) which forces "slaves" rely on the source that advertises the best accuracy. According to the PTP standard, a client with 2 sources, one of which suddenly announces itself as more accurate than the other is required to rely on that source, no matter what other information it may have. The EUREX clock that failed instructed PTP clients to stay with its obviously incorrect new time, and they obeyed. BMC, thus, introduces a single point of failure and subtracts intelligence from the "slaves". Generally, sources don't even try to estimate network delays to the client making the accuracy estimate useless. The next enterprise PTP standard attempts to walk back from BMC but it is deeply embedded in the standard and in more traditional implementations. TimeKeeper will accept time from multiple sources, using either or both protocols, cross check them, and failover down a list provided by IT staff who should know which sources are closer to the client and are better quality. Sophisticated filtering and smoothing, smart error detecting, end-to-end information - all are parts of making TimeKeeper more reliable[10, 9, 8].

4.7. **Measuring accuracy.** Measurement of clock accuracy is a common source of error: both because some clock synchronization software is highly optimistic about how good it is and fundamental properties of timing. For example, calculating one way delay from round trip time involves attempting to solve a single equation with two unknown variables: we know round trip time is the sum of the two one way delays. There have been countless hopeless efforts to move this immovable mathematical obstacle recorded in many academic papers and even patent filings. Fast, symmetric network equipment is the solution. It is also important to remember that if the clock sync software "knew" exactly how far off it was, there would be no reason not to have perfect time. There is not even a standard for synchronization error measurement. TimeKeeper's method of, in real-time, monitoring multiple sources, improves this situation a great deal and TimeKeeper's accuracy estimate is designed to be a pessimistic estimate of how well the client software is able to track the source clock signal. If there is a single clock source TimeKeeper will look for internal consistency, but a stable, incorrect, single time source is impossible for client software to detect.

REFERENCES

[1] David Allan et al. *US Patent 5,274,545: Device and method for providing accurate time and/or frequency*. Washington, DC, 1992.

[2] Bevin B. *Memory Performance in a Nutshell*. https://software.intel.com/en-us/articles/memory-performance-in-a-nutshell. 2016.

[3] Charles Babcock. *NTP's Fate Hinges On 'Father Time*. https://www.informationweek.com/it-life/ntps-fate-hinges-on-father-time/d/d-id/1319432.

[4] Eric Brewer. *Spanner, TrueTime and the CAP Theorem*. Tech. rep. 2017.

[5] Bruce Byfield. *A rift in the NTP world*. https://lwn.net/Articles/713901.

[6] James C. Corbett et al. "Spanner: Google&Rsquo;s Globally Distributed Database". In: *ACM Trans. Comput. Syst.* 31.3 (Aug. 2013), 8:1–8:22. ISSN: 0734-2071. DOI: 10.1145/2491245. URL: http://doi.acm.org/10.1145/2491245.

[7] Council of European Union. *Commission Delegated Regulation (EU) 2015/574*. http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.L_.2017.087.01.0148.01.ENG&toc=OJ:L:2017:087:TOC. 2017.

[8] Cort Dougan and Victor Yodaiken. *US Patent 9348321: Method, time consumer system, and computer program product for maintaining accurate time on an ideal clock*. Austin, TX, 2013.

[9] Cort Dougan and Victor Yodaiken. *US Patent 9671761: Method, time consumer system, and computer program product for maintaining accurate time on an ideal clock*. Austin, TX, 2016.

[10] Cort Dougan and Victor Yodaiken. *US Patent 9756153: Method for improving accuracy in computation of one-way transfer time for network time synchronization*. Austin, TX, May 2012.

[11] Pedro Estrela and Lodewijk Bonebakker. "Challenges deploying PTPv2 in a global financial company". In: ISPCS (Sept. 2012), pp. 1–6.

[12] Yilong Geng et al. "Exploiting a Natural Network Effect for Scalable, Fine-grained Clock Synchronization". In: *15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18)*. Renton, WA: USENIX Association, 2018, pp. 81–94. ISBN: 978-1-931971-43-0. URL: https://www.usenix.org/conference/nsdi18/presentation/geng.

[13] Allen R. Goldstein. *Time Distribution Alternatives for the Smart Grid Workshop Report*. Tech. rep. 1500-12. 2017.

[14] Kang Lee and John Eidson. "IEEE-1588 Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems". In: *In 34 th Annual Precise Time and Time Interval (PTTI) Meeting*. 2002, pp. 98–105.

[15] M. Lewis. *Flash Boys: A Wall Street Revolt*. W. W. Norton, 2014. ISBN: 9780393244670. URL: https://books.google.com/books?id=AarfAgAAQBAJ.

[16] Barbara Liskov. "Practical Uses of Synchronized Clocks in Distributed Systems". In: *Proceedings of the Tenth Annual ACM Symposium on Principles of Distributed Computing*. PODC '91. Montreal, Quebec, Canada: ACM, 1991, pp. 1–9. ISBN: 0-89791-439-2. DOI: 10.1145/112600.112601. URL: http://doi.acm.org/10.1145/112600.112601.

[17] *Network Time Protocol (Version 3) Specification, Implementation and Analysis*. RFC 1305. Mar. 1992. DOI: 10.17487/RFC1305. URL: https://rfc-editor.org/rfc/rfc1305.txt.

[18] Matt Sherer. *Comparing NTP and PTP*. https://www.fsmlabs.com/news/2015/03/12/ptpvsntp.html. 2015.

[19] Robert H. Thomas. "A Majority Consensus Approach to Concurrency Control for Multiple Copy Databases". In: *ACM Trans. Database Syst.* 4.2 (June 1979), pp. 180–209. ISSN: 0362-5915. DOI: 10.1145/320071.320076. URL: http://doi.acm.org/10.1145/320071.320076.

AUSTIN TEXAS.
E-mail address: yodaiken@fsmlabs.com