1 Introduction

At the 19 July 2023 IEEE P3335 Plenary, it emerged that the "Data Center" Use-Case should encompass large-scale distributed systems and databases, including large facilities thousands of kilometers apart, where very high precision timestamps (with a histogram peak width of around twenty nanoseconds) would be used to deduce the time order in which events happen[1].

Distributed systems and databases were studied extensively some 45 years ago (in the 1970s) and the performance bounds were derived and proven mathematically. The classic and definitive paper is Lamport's "Time, Clocks, and the Ordering of Events in a Distributed System" [Lamp78], 8 pages. Although short and very clearly written, it is still a dense mathematical paper, so a focused tutorial summary is provided here.

Note that random and malicious (byzantine) faults are not handled here; see [Lamp82] for the details of handling of faults, and how many end-to-end messages are required to overcome a specified number of faults

Nor are safety-critical interlocked-message protocols addressed.

2 Definitions

In the following, definitions and results from *Lamport* are quoted to illuminate the fundamental constraints governing the Data-Center Use-Case:

"A *distributed system* consists of a collection of distinct processes which are spatially separated, and which communicate with one another by exchanging messages. ... A single computer can also be viewed as a distributed system in which the central control unit, the memory units, and the input-output channels are separate processes."

The word *message* is used loosely, and includes hardware signals.

"A system is *distributed* if the message transmission delay is not negligible compared to the time between events in a single process."

Here, an *event* takes no time, being an instant, and no two events can happen at once – one event is either before or after the other, and never simultaneous.

[1] It's unclear if the mean or median is assumed to be zero; this will be defined. This center will wander about over tens to thousands of seconds, as captured in ADEV data on the GPS receiver.

The underlying assumed model is that of a finite state machine, which has states and moves between those states instantly upon arrival of this or that event[2]. Events can be anything that has zero duration, including the expiration of a timer.

If one has a collection of finite-state machines communicating using messages suffering a finite transport delay, the resulting composite cannot be treated as a larger state machine because *finite delay* conflicts with *instantaneous*, so this composite cannot possess an overall state.

3 Special Relativity

In systems that fit entirely within a sphere a few meters in diameter, the transport delays are mostly due to the limitations of electronic components and the like. But when the messages travel kilometers, the speed of light defines the minimum possible transport delay. According to Einstein's *Special Theory of Relativity*, information (including those messages) cannot travel faster than the speed of light in vacuum.

Special Relativity also holds that it is often impossible to know which event in physically separated process happened first because the knowledge of those events is still in flight, at least one announcing message not having been received yet. And depending on the location of the viewer, different orderings will be seen, all valid. So, there will be periods of time where some processes have beliefs contrary to fact in that other processes have changed state, but the news has not yet reached all processes.

Note that because this is a fundamental physics constraint, it cannot be evaded by software, however clever.

In practice, the messages are carried in fused-silica (glass) optical communications fiber, where information travels at about two thirds of lightspeed[3].

A numerical example is in order. The transport delay through ten kilometers of glass optical communications fiber is $10^4/(2\ 10^8) = 500\ \mu s$, which well exceeds 20 ns, the width of the PTPv2.1 histogram peak.

---

[2] Markov Sequences or Chains are generated by a finite state machine effectively driven by random events of specified probability. < https://en.wikipedia.org/wiki/Markov_chain>

[3] Interesting diversion: High-Speed Traders use microwave beams in air to outrun messages carried on fiber. A typical path would be between Chicago and New York City. See "Flash Boys – A Wall Street Revolt", Michael Lewis, Norton 2014 for the story, and "Relativistic statistical arbitrage", A. D. Wissner-Gross and C. E. Freer (both of MIT), Physical Review E 82, 056104, 2010 for the deep analysis of how and why it works.

4 Partial and Total Ordering

Total ordering is where the events are in strict time order. This is needed if for instance database *ACID* transaction properties (described later) are to be achieved, which is necessary to guarantee that a payment is made exactly once, that a resource cannot be double-booked, and so on, as discussed later.

Partial ordering is where the time ordering is approximate, most commonly where there is a moving time window within which ordering may not be total (exact), and so ACID properties cannot be guaranteed.

Lamport gives the precise conditions needed to ensure exact Total Order.

5 Transaction Processing and ACID Properties

The following are copied from IBM's online documentation[4]. Total Ordering of events is required for ACID properties to be guaranteed.

5.1 Atomicity

All changes to data are performed as if they are a single operation. That is, all the changes are performed, or none of them are.

For example, in an application that transfers funds from one account to another, the atomicity property ensures that, if a debit is made successfully from one account, the corresponding credit is made to the other account.

5.2 Consistency

Data is in a consistent state when a transaction starts and when it ends.

For example, in an application that transfers funds from one account to another, the consistency property ensures that the total value of funds in both the accounts is the same at the start and end of each transaction.

5.3 Isolation

The intermediate state of a transaction is invisible to other transactions. As a result, transactions that run concurrently appear to be serialized.

For example, in an application that transfers funds from one account to another, the isolation property ensures that another transaction sees the transferred funds in one account or the other, but not in both, nor in neither.

5.4 Durability

After a transaction successfully completes, changes to data persist and are not undone, even in the event of a system failure.

[4] .<https://www.ibm.com/docs/en/cics-ts/5.4?topic=processing-acid-properties-transactions>

For example, in an application that transfers funds from one account to another, the durability property ensures that the changes made to each account will not be reversed.

6 Summary and Conclusions

AI and Google PageRank algorithms are the application of linear algebra to immense matrices, basically computing cross-correlations over inherently noisy data, so one would think that Partial Ordering suffices, unless the level of misordering of events (arrival of messages carrying packages of data) is quite large. The consequence of time-order errors will ordinarily make a small addition to the already large inherent noise levels inherent in the data.

This observation leads to the question if this P3335 Data-Center Use-Case requires total ordering, or is partial ordering sufficient? In other words, is the need all or nothing, or more likely, only a tiny fraction of messages exchanged must achieve Total Order (and thus incur multiple end-to-end transport delays), allowing overall system performance to be dominated by timestamp error distributions, not round-trip latency across the entire distributed system.

7 Notional Data-Center Facility Characteristics

There are at least two GPS Receivers (and associated antennas) that are sufficiently dispersed physically that no single lightning bolt can destroy all of them.

The GPS receivers all feed time to PTP-enabled network switch via optical fiber, for general EMI immunity, and to contain the destructive effect of a lightning strike on a GPS antenna and its receiver.

The network switch may convert PTP traffic from fiber to copper, or may feed added network switches via optical fiber, eventually converting to PTP via copper where needed.

The copper PTP links feed a PCIe Time Card mounted in each server computer needing nanosecond time. Otherwise, a PTP-enabled (perhaps PCIe) ethernet NIC may be used.

It may be necessary or at least useful to have a lightly-loaded isolated ethernet “Realtime” LAN used only for carriage of latency-critical traffic, where the maximum packet size is a few hundred bytes. A different LAN would be used for bulk carriage of data with maximum-size packets, including 9 Kbyte jumbo packets. In short, the first LAN is optimized for low and predictable latency and latency jitter, the second LAN for throughput alone.

8 References

[Gray92] "Transaction Processing: Concepts and Techniques, 1st Edition", by Jim Gray and Andreas Reuter, Morgan Kaufmann 1992, 1128 pages.

[Lamp82] "The Byzantine Generals Problem", Leslie Lamport, Marshall Pease, and Robert Shostak; ACM Transactions on Programming Languages and Systems 4, 3 (July 1982), 382-401.

[Lamp78] "Time, Clocks, and the Ordering of Events in a Distributed System" by Leslie Lamport, Communications of the ACM, July 1978, Volume 21, Number 7, pages 558-565. This is paper 27 in *My Collected Works* in Lamport's website[5]. The associated discussion is illuminating.

[Steen17] "Distributed Systems 3rd Edition", by Maarten van Steen and Andrew S. Tanenbaum, CreateSpace 2017, 596 pages.

9 Acronyms

ACID = {Atomicity, Consistency, Isolation, and Durability}, **ADEV** = Allen Deviation, **AI** = Artificial Intelligence, **Kbyte** = Kilobyte, **LAN** = Local Area Network, **NIC** = Network Interface Card, **ns** = nanosecond, **µs** = microsecond, **PCIe** = Peripheral Component Interconnect Express, **PTP** = Precision Time Protocol (IEEE 1588)

[5] .< https://lamport.azurewebsites.net/?from=https://research.microsoft.com/users/lamport/&type=exact>