HSLU Hochschule Luzern

Verteilte Systeme und Komponenten

Koordination verteilter Systeme

Martin Bättig
(basierend auf Material von Roger Diehl)

Letzte Aktualisierung: 14. Dezember 2022

FH Zentralschweiz

Inhalt

- Physische Zeit
- Logische Zeit
- Lamport-Zeitstempel
- Vektor-Zeitstempel

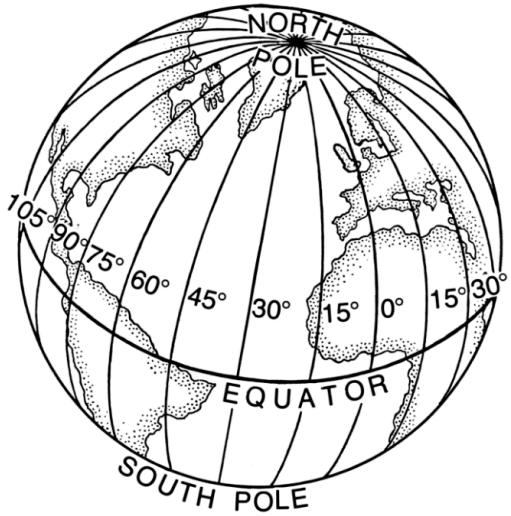
Lernziele

- Sie kennen zwei verschiedene Algorithmen zur Synchronisation von physischen Uhren.
- Sie wissen was eine logische Uhr ist.
- Sie kennen die Happened-Before-Relation.
- Sie kennen die Algorithmen des Lamport-Zeitstempels und des Vektor-Zeitstempels zur Synchronisation von logischen Uhren.
- Sie können die Algorithmen zur Synchronisation von logischen Uhren in eigenen Programmen implementieren.

Physische Zeit

Bedeutung von Zeit

- Bestimmung der Zeit und deren Messung unverzichtbar zur Koordination menschlicher Aktivitäten.
- Koordination erreicht durch Synchronisation von Zeitmessern (Uhren).
- Synchronisation der Uhren mittels Kirchturmuhr, Telegraphie, Radio,…
- Uhrensynchronisation ermöglichte in der Schifffahrt erst die Längengradbestimmung.
- Die Existenz einer globalen Zeit haben wir verinnerlicht.

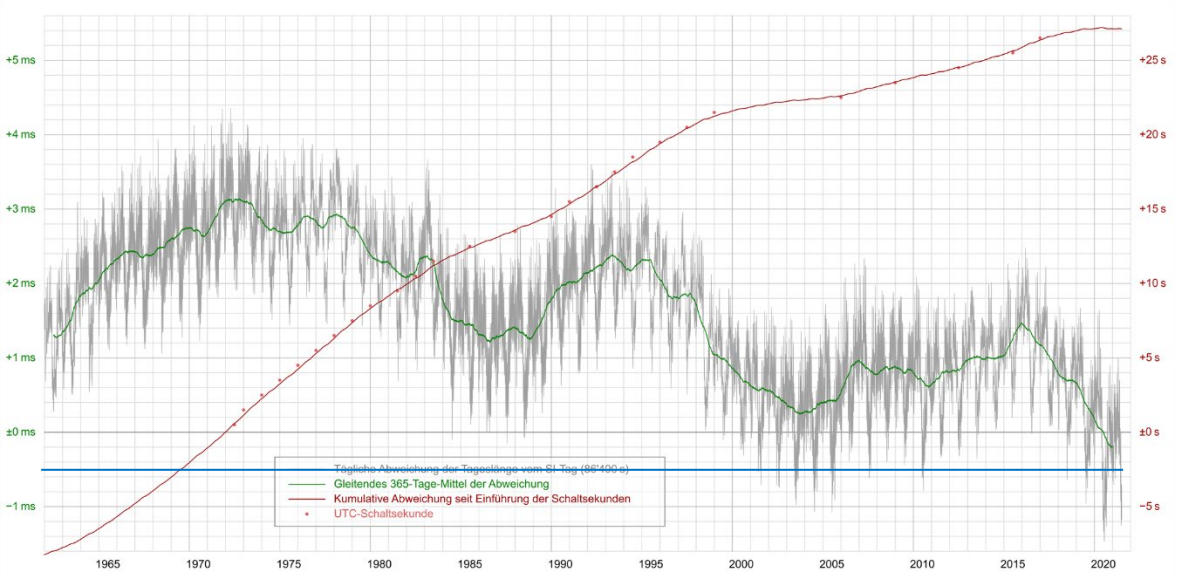


Astronomische Uhr
Zytturm Zug

Was ist Zeit?

- Im 16. Jahrhundert - Einführung Gregorianischer Kalender.
- Im 17. Jahrhundert - Durchgang der Sonne im Zenit.
  - 1 Sonnentag = Zeit zwischen zwei Zenit Durchgängen.
  - 1 Sonnensekunde = 1/86400 eines Sonnentages.
- TAI - International Atomic Time stellt seit 1.1.1958 Anzahl Ticks der Cäsium 133-Uhren zur Verfügung.

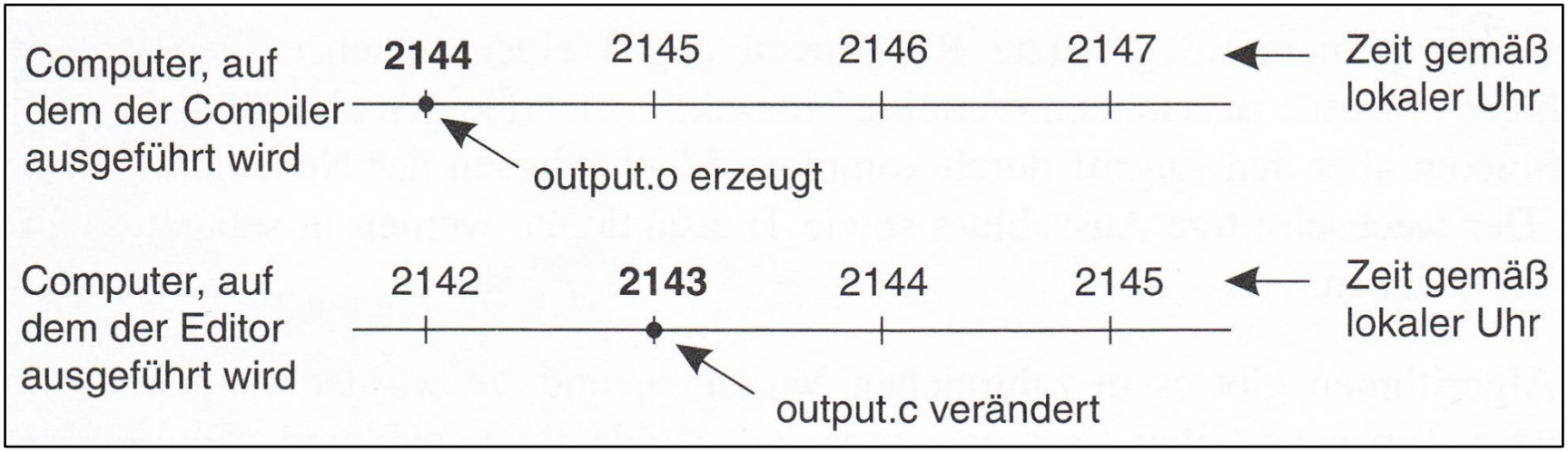
- seit Einführung sind 86400 TAI Sek. im Mittel 2ms kürzer als ein Sonnentag.



https://www.timeanddate.com/time/leapseconds.html

Was passieren kann…

- Falls es keine globale Einigung auf die Zeit gibt ist folgendes Szenario denkbar:



- Konsequenz: `output.c` wurde scheinbar zu einem früheren Zeitpunkt erstellt, deshalb wird nicht neu kompiliert!
  - Es entsteht eine Mischung aus alten und neuen Dateien.

Ist es möglich alle Uhren in einem verteilten System zu synchronisieren?

Voraussetzung für Uhren-Synchronisierung

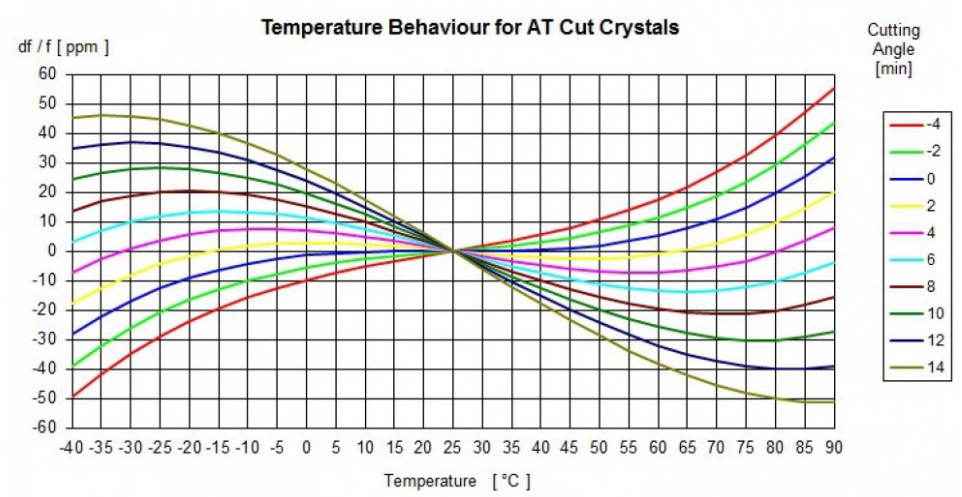
Timer: Schaltung in Computern, welche die Zeit verwaltet.

- Quarzkristall unter Spannung schwingt mit bestimmter
- Frequenz.
- Zählerregister zählen Schwingungen mit und erzeugen Interrupts in bestimmten Intervallen.

Tick: Durch den Timer erzeugter Interrupt.

Uhrasymmetrie: Unterschiede von Zeitwerten verschiedener Uhren, auch wenn diese ursprüngliche synchronisiert waren.

- Zeitwerte laufen auseinander, weil Quarzkristalle mit unterschiedlicher Qualität verwendet werden und deshalb mit unterschiedlichen Frequenzen schwingen.



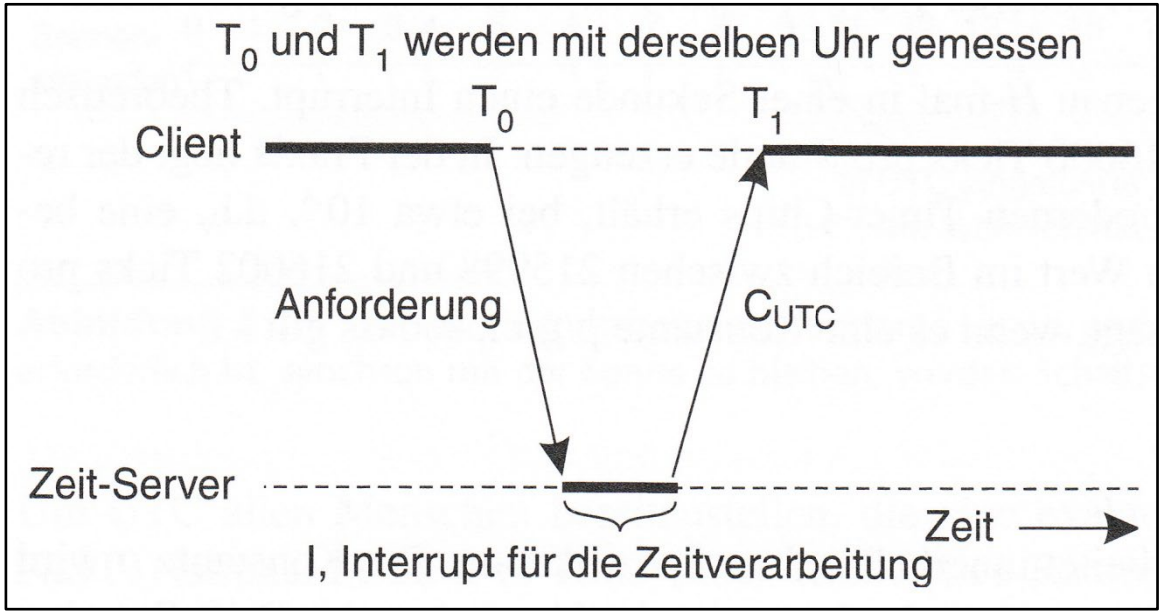
Quelle: https://www.iqdfrequencyproducts.de

Algorithmus von Cristian

Zeitserver: Maschine mit Zeitzeichenempfänger*, mit diesem Server werden alle anderen Maschinen synchronisiert.

– Zeitzeichensender sendet am Anfang jeder UTC-Sekunde einen kurzen Impuls.

– UTC – Universal Coordinated Time: Zeitmessung in Beziehung mit dem Sonnenstand mit Schaltsekunden.



Passives System

- z.B. Langwellensender DCF77: https://de.wikipedia.org/wiki/DCF77

F. Cristian: *Probabilistic clock synchronization.* In: *Distributed Computing.* Volume 3, Issue 3, 1989, S. 146–158.

Algorithmus von Cristian

1. Client P erfragt die Zeit von Zeit-Server S zum Zeitpunkt $t_{0.}$
2. Die Anfrage wird von S verarbeitet – dies benötigt eine Zeitspanne I.
3. Die Antwort $C_{UTC}(t_1)$ wird von P zum Zeitpunkt t_1 empfangen.
4. P wird auf die Zeit $C_{UTC}(t_1)$ + RTT/2 gesetzt, d.h. die vom Server gemeldete Zeit plus die Rücklaufzeit des Pakets.
   - die Round Trip Time (RTT) wird dabei berechnet durch RTT = t_1 - $t_{0.}$
   - ist die Zeitspanne I bekannt, kann die Berechnung verbessert werden RTT = t_1 - t_0 - I.
5. Für genauere Werte wird die Laufzeit öfters gemessen, Messungen ausserhalb eines Bereiches werden verworfen und eine Mittelung der restlichen Werte durchgeführt.

Algorithmus von Cristian – Probleme

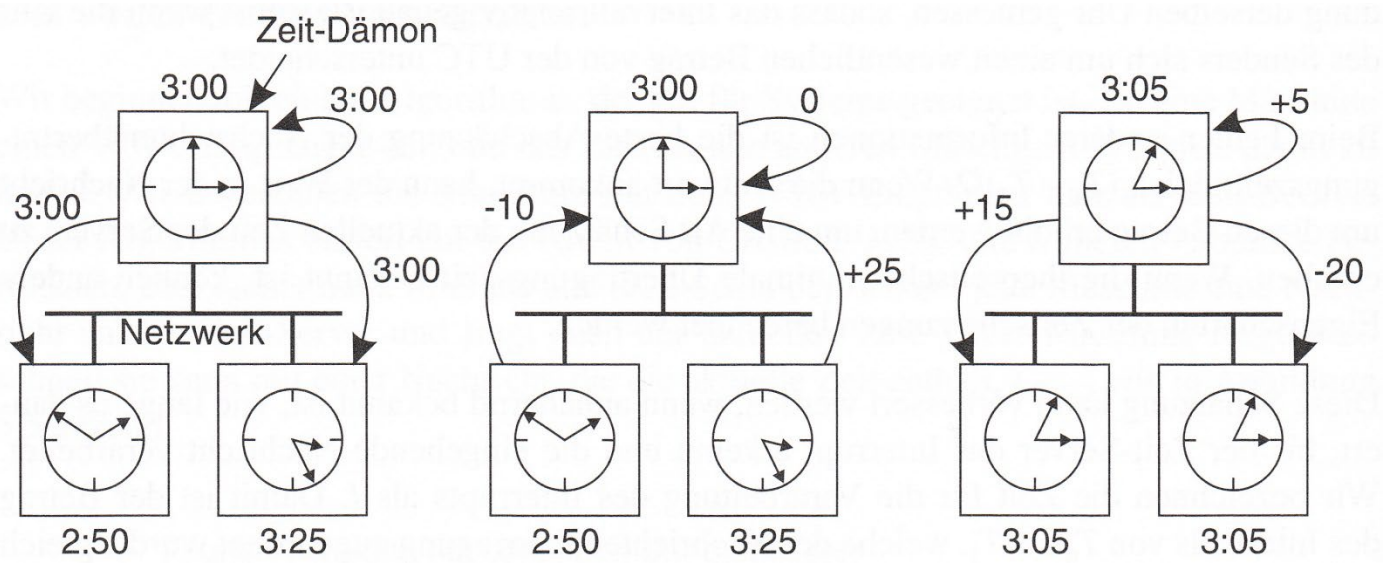
Grosses Problem: Zeit kann nicht rückwärts laufen.

- Zeit vom Zeitserver liegt in der Vergangenheit der lokalen Zeit.
- Zeit kann nicht einfach zurück gedreht werden, da inkonsistente Zustände im System entstehen könnten.
- **Lösung:** Verlangsamung der lokalen Zeit, bis Zeitdifferenz ausgeglichen.

Kleines Problem: Antwort des Zeitservers braucht Zeit.

- Laufzeit der Anfrage kann nicht genau bestimmt werden, abhängig von Netzwerklast.
- Kompensation durch mehrfache Messung der Dauer der Anfrage und Adaption des vom Zeitservers gelieferten Wert.

Berkeley-Algorithmus



Aktives System

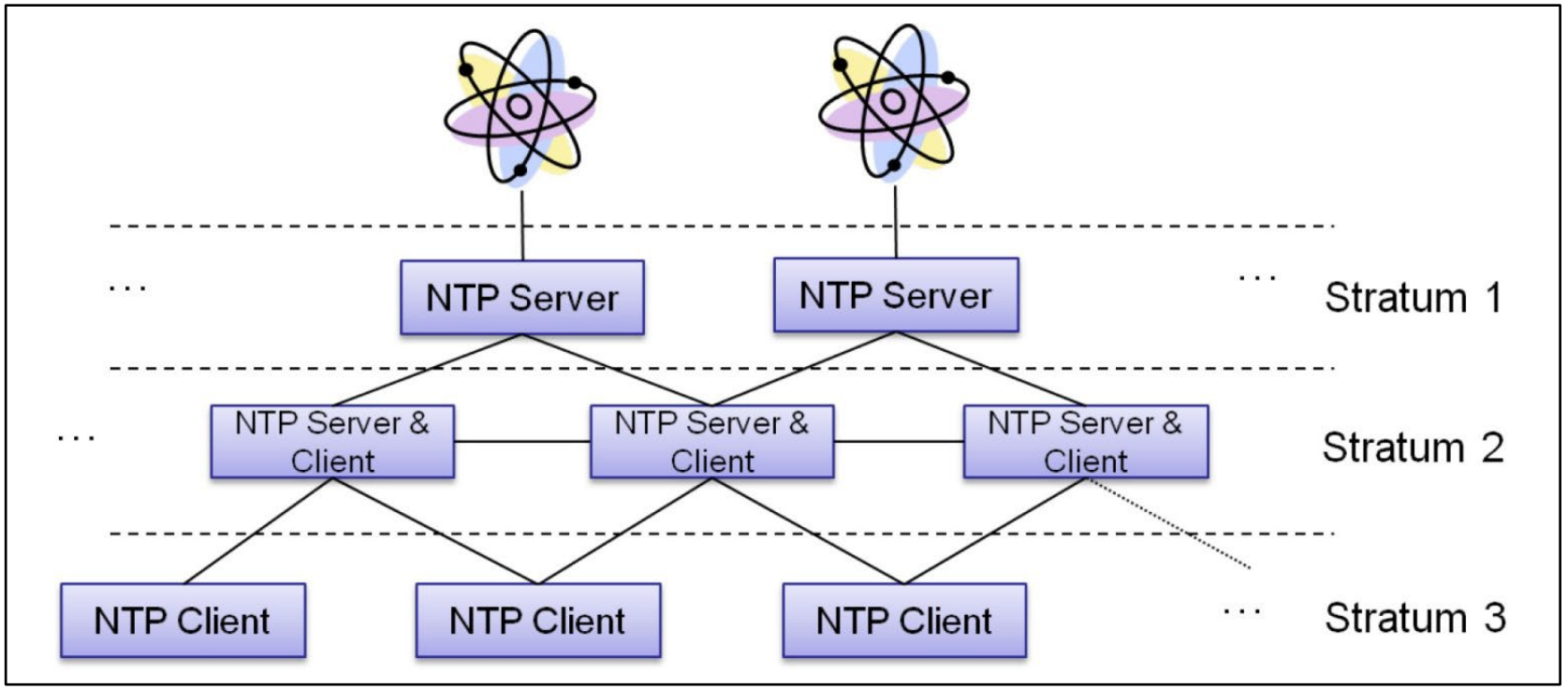
- Keine Maschine hat einen Zeitzeichenempfänger.
- Der Zeitserver (Zeit-Dämon) fragt in regelmässigen Abständen die lokale Zeit von allen teilnehmenden Clients ab.
- Basierend auf den Antworten berechnet der Zeitserver eine Durchschnittszeit und weist alle Maschinen an, ihre Uhren der neuen Zeit anzupassen.

Network Time Protocol - NTP

- Entwickelt seit 1982 (NTP v1, RFC 1059) unter Leitung von David Mills; Aktuelle Version NTP v4, seit 1994.
- Zweck: Synchronisierung von Rechneruhren im Internet.
- NTP-Dämon auf fast allen Rechnerplattformen verfügbar, von PCs bis Crays; Unix, Windows, VMS, eingebettete Systeme.
- Erreichbare Genauigkeiten von ca. 10 ms in WANs und kleiner als 1ms in LANs.
- Fehlertolerant.
- Ausführliche Informationen zu NTP:
  - http://www.ntp.org ("Offizielle" NTP-Homepage).
  - https://www.eecis.udel.edu/~mills (Homepage David Mills).
  - http://www.ntpclient.com (Infos zu NTP Client Software).

NTP - Struktur

- **Stratum 1:** primärer Zeitgeber, über Funk oder Standleitungen an amtliche Zeitstandards angebunden.
- **Stratum >1:** synchronisiert mit Zeitgeber des Stratums N - 1.
- Stratum kann dynamisch wechseln, z.B. bei Unterhalt oder Ausfall der Verbindung.



NTP - Datenpaket

NTPv4: https://tools.ietf.org/html/rfc5905

Seconds (32-bit): Anzahl Sekunden seit 1.1.1900

Quelle: https://www.meinberg.de/german/info/ntp-packet.htm

Prinzipieller Ablauf

1. Client sendet eine NTP-Message an den Timeserver.
2. Server verarbeitet Paket (passt IP-Addressen, Timestamps, weitere Felder an).
3. Server sendet Paket zurück.
4. Client hat nun vier Zeitstempel (t1-t4) und leitet davon Offset und Delay ab:

Client	Connection	Server
Zeitstempel 1	⇨	Zeitstempel 2
Zeitstempel 3	⇦	Zeitstempel 4

➡

Offset =	$\frac{(t2 - t1) + (t3 - t4)}{2}$
Delay =	$(t4 - t1) - (t2 - t3)$

- **Offset:** Zeitdifferenz der Rechneruhren (gemittelt).
  Um diesen Wert wird die Zeit geändert, falls die Qualität der Messung gut ist.
- **Delay:** Zeit während der das Paket unterwegs war.
  Mass für die Qualität. Ggf. werden mehrere Pakete versandt und dasjenige mit dem geringsten Delay der letzten acht Pakete verwendet.

Logische Zeit

Logische Zeit

- 1978 zeigte Leslie Lamport (*), dass es ausreichend ist, wenn sich alle Maschinen über dieselbe Zeit einig sind.
- Eine Übereinstimmung mit der Zeit ausserhalb des Systems ist nicht notwendig (keine physische Zeit nötig).
- Logische Zeit findet vor allem in Bereichen Anwendung, in denen Kausalität und Verlässlichkeit eine grosse Rolle spielen.
- Allerdings sind die Verfahren zur Synchronisation von logischen Uhren in grossen Systemen im Allgemeinen ineffizient.

(*) Leslie Lamport: US-amerikanischer Mathematiker, Informatiker und Programmierer. 2013 erhielt er den Turing Award für seine Beiträge zur Theorie und Praxis verteilter und nebenläufiger Systeme.

Happened-Before-Relation von Lamport

- Der Ausdruck a → b wird gelesen als «a passiert vor b»
  - bedeutet, dass sich alle Prozesse einig sind, dass
  - zuerst das Ereignis a stattfindet und dann das Ereignis b.

- Direkte Beobachtung der Relation in zwei Situationen:
  1. wenn a und b Ereignisse im selben Prozess sind, und a vor b auftritt, gilt a → b.
  2. wenn a das Senden einer Nachricht bei einem Prozess und b das Empfangen derselben Nachricht bei einem anderen Prozess ist, dann gilt a → b.

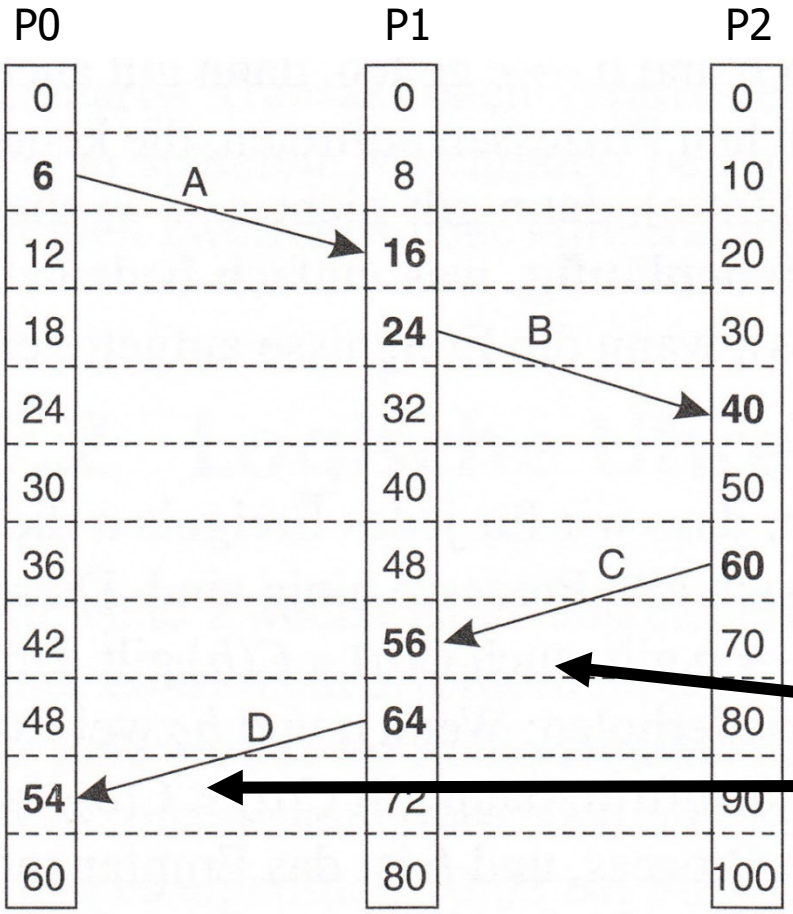
- Zwei Ereignisse a ≠ b sind kausal unabhängig, geschrieben als a || b, wenn weder a → b noch b → a sind

- Happened-Before-Relation ist transitiv: Wenn a → b und b → c gelten, dann gilt auch a → c

Lamport-Zeitstempel

Ausgangslage

Jede Maschine hat eine eigene Zeit mit konstanten aber unterschiedlichen Geschwindigkeiten.



Prozess P0 sendet Prozess P1 Nachricht A

Prozess P1 sendet Prozess P2 Nachricht B

Prozess P2 sendet Prozess P1 Nachricht C

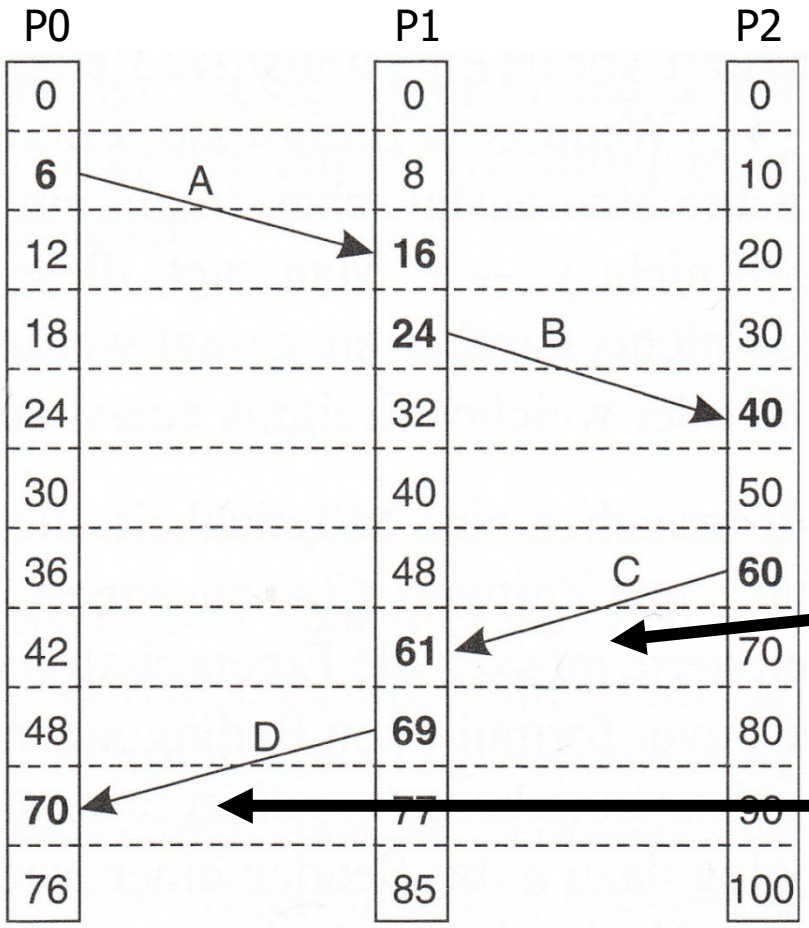
Nachricht C und D kommen scheinbar «früher» an als sie gesendet wurden

Lamport-Zeitstempel

- Ein Prozess sendet eine Nachricht mit Zeitstempel (eigene Zeit) an einen anderen Prozess.
- Einem Ereignis a wird ein Zeitwert C(a) zugeordnet.
  - alle Prozesse sind sich über den Zeitwert einig.
  - wenn a → b gilt auch C(a) < C(b).
- Ein Prozess sendet eine Nachricht mit Zeitstempel a (eigene Zeit) an einen anderen Prozess, welcher die Nachricht zur eigenen Zeit b empfängt, dann müssen C(a) und C(b) so zugewiesen werden, dass C(a) < C(b) ist.
- Die Zeit C muss immer vorwärts laufen.
  - ansteigende Werte.
- Korrekturen können durch Addition von positiven Werten vorgenommen werden.

Lösung

Zwischen zwei Ereignissen muss die lokale Zeit mindestens einmal ticken – empfangene Zeit + 1.



Senden ist vor Empfangen der Nachricht A - OK

Senden ist vor Empfangen der Nachricht B - OK

Empfangen (t=56) ist vor dem Senden der Nachricht C – Korrektur bei Prozess P1

Empfangen (t=54) ist vor dem Senden der Nachricht C – Korrektur bei Prozess P0

Quelle: https://de.wikipedia.org/wiki/Lamport-Uhr

Lamport-Zeitstempel zusätzliche Forderung

Zwei Ereignisse dürfen nie zu genau der selben (logischen) Zeit auftreten.

Lösung: Zeitstempel um Prozessnummer ergänzen.

Damit kann allen Ereignissen in einem verteilten System eine Zeit zugewiesen werden, die folgenden Bedingungen erfüllt:

1. wenn a im selben Prozess vor b auftritt, gilt $C(a) < C(b)$.
2. wenn a und b das Senden und Empfangen einer Nachricht darstellen, gilt $C(a) < C(b)$.
3. für alle anderen Ereignisse a und b, gilt $C(a) \neq C(b)$.

Lamport-Zeit – Eigenschaften

- Lamports Uhren erfüllen die Uhrenbedingung: $a \rightarrow b \Rightarrow C(a) < C(b)$.
  Wenn Ereignis a vor Ereignis b stattfindet, dann ist der Zeitstempel von C(a) kleiner als der von C(b).
- Die logischen Lamport-Zeitstempel definieren daher eine partielle Ordnung auf der Menge der Ereignisse, die den kausalen Zusammenhang zwischen Ereignissen erhält.
- Ergänzung zu einer totalen Ordnung ist wieder möglich.

Einschränkung: Anhand der Zeitstempel lässt sich nicht immer sicher sagen, ob zwei Ereignisse kausal voneinander abhängen.

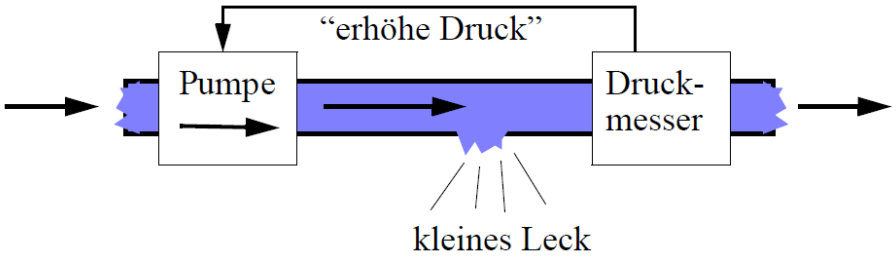
- hierfür müsste auch die Umkehrung der Uhrenbedingung gelten, aber es gilt lediglich $C(a) < C(b) \Rightarrow a \rightarrow b \vee a \,||\, b$

Vektor-Zeitstempel

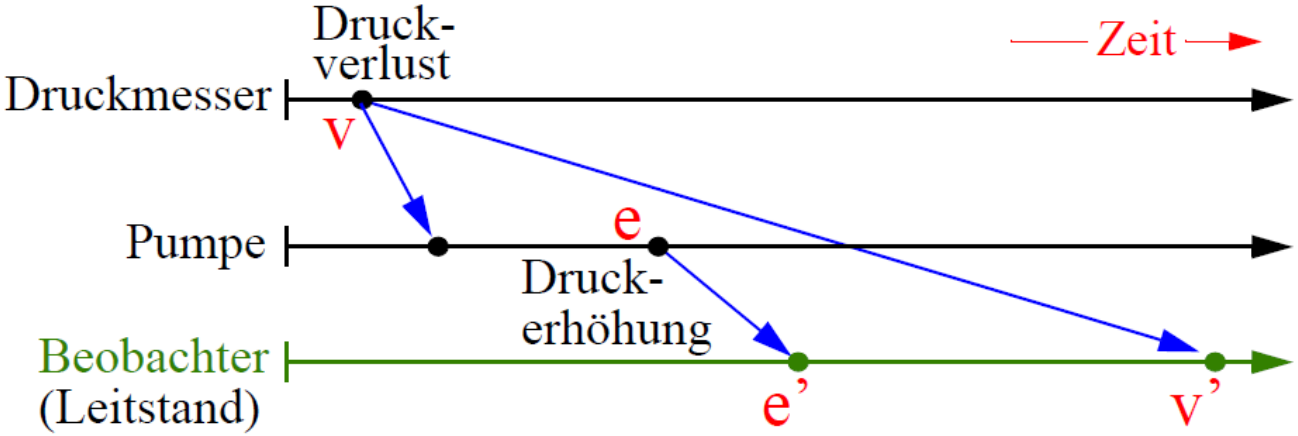
Beispiel: (nicht) kausaltreue Beobachtungen

Gewünscht: Eine Ursache stets vor ihrer (u.U. indirekter) Wirkung beobachten.

Was passiert ist:



Eingehende Nachrichten:



Falsche Schlussfolgerung des Beobachters:

Es erhöhte sich der Druck (aufgrund einer Aktivität der Pumpe), es kam zu einem Leck, was durch den abfallenden Druck angezeigt wird.

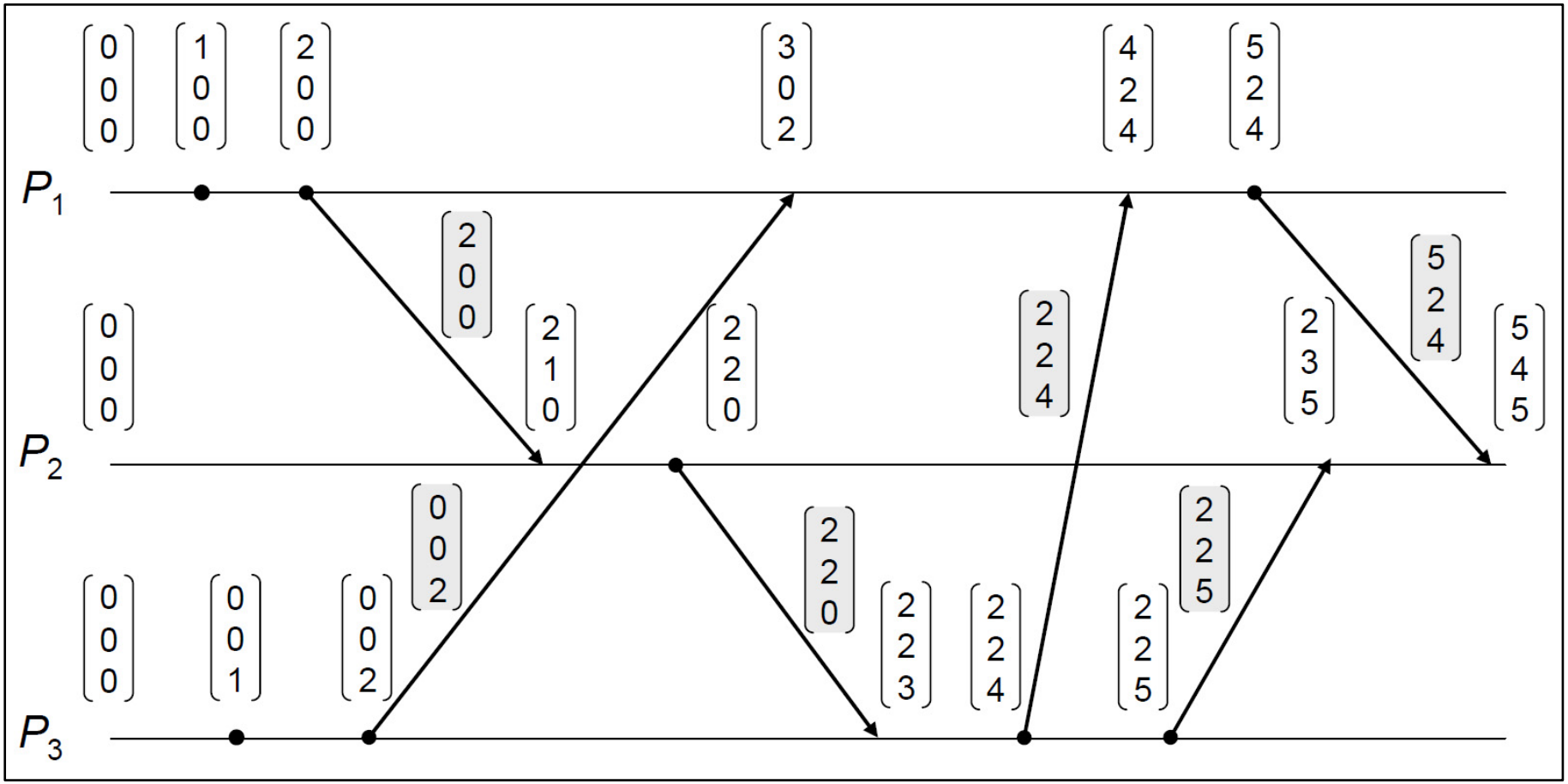
Definition

- Ein Vektor-Zeitstempel VT(a), der einem Ereignis a zugewiesen wurde, hat die Eigenschaft, dass Ereignis a dem Ereignis b kausal vorausgeht, wenn VT(a) < VT(b) für ein Ereignis b gilt.
- Jeder Prozess P_i besitzt einen Vektor V_i, der für jeden Prozess im System die Anzahl der Ereignisse enthält mit den Eigenschaften:
  - $V_i[i]$ ist die Anzahl der Ereignisse, die bisher in P_i aufgetreten sind
  - wenn gilt $V_i[j] = k$, erkennt P_i, dass in P_j k Ereignisse aufgetreten sind.
- Der Vektor V_i wird den gesendeten Nachrichten mitgegeben.

Algorithmus Vektor Zeitstempel

- Jeder Prozess P_i hält einen Vektor V_i bestehend aus n Zählern (n = Anzahl der Prozesse im System).
- Initial ist der Vektor Zeitstempel jedes Prozesses der Nullvektor.
- Tritt bei Prozess P_i ein Ereignis auf, so inkrementiert er die i-te Komponente seines Vektor.
- Sendet P_i eine Nachricht, so wird die neue Version von V_i mitgeschickt.
- Empfängt P_i eine Nachricht mit Vektor Zeitstempel VT, so bildet er das komponentenweise Maximum von der neuen Version von V_i und von VT.

Beispiel Vektor-Uhren



grau: gesendeter Vektor Zeitstempel

Informationen des Vektor-Zeitstempel

Der Vektor-Zeitstempel in der Nachricht informiert Empfänger über

- die Anzahl Ereignisse die in P_i aufgetreten sind,
- wie viele Ereignisse in anderen Prozessen der Nachricht vorausgegangen sind,
- wie viele vorangegangene Ereignisse möglicherweise kausal abhängig sind.

Kausaler Zusammenhang zwischen zwei Ereignissen

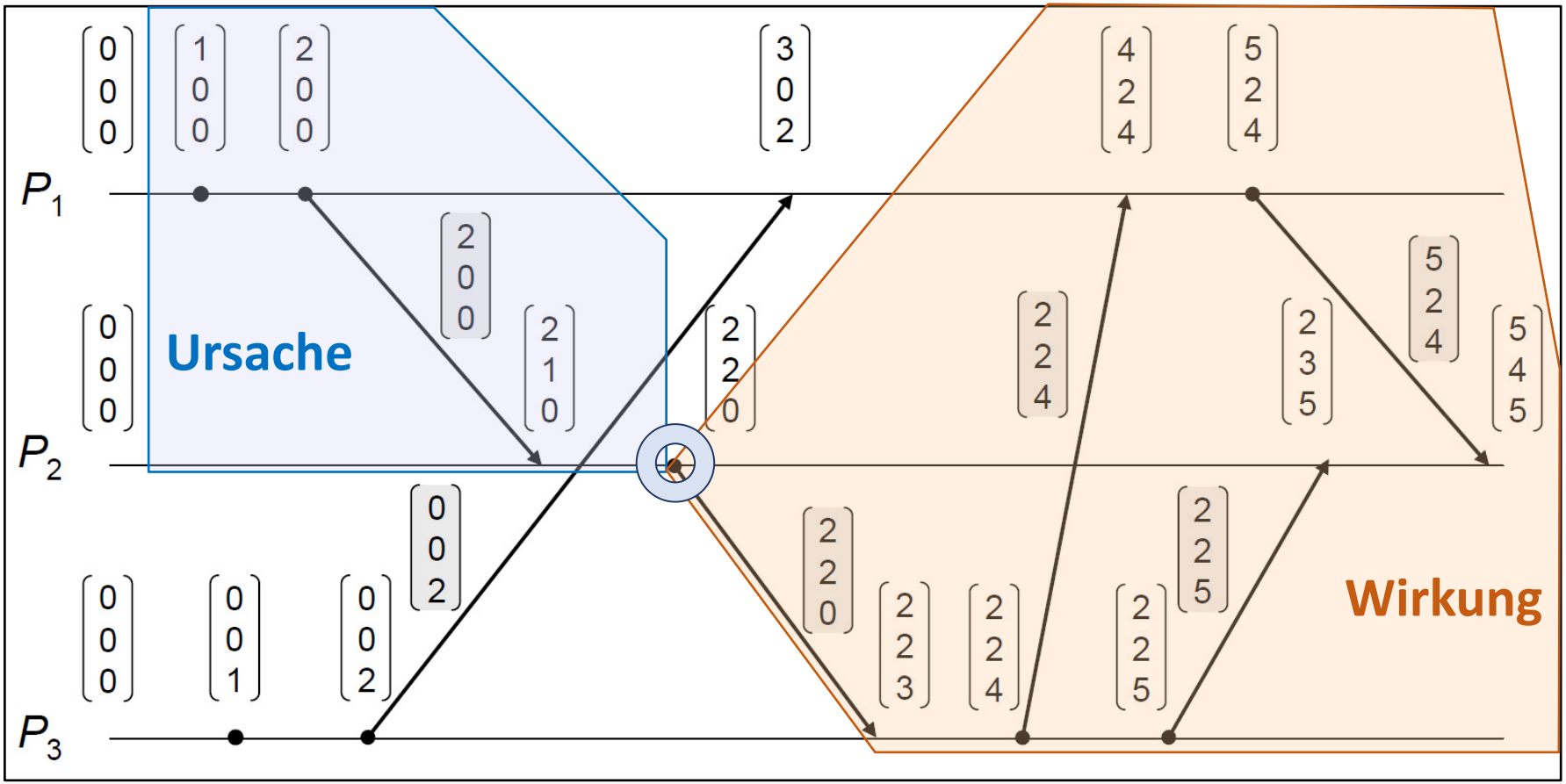
Ereignis A ist eine Ursache von Ereignis B:

- wenn der Zähler für jeden Prozess im Zeitstempel VT(A) kleiner oder gleich dem Zähler im Zeitstempel VT(B) für den korrespondierenden Prozess
- und für mindestens einen dieser Zähler kleiner ist.

Beispiele:

- $\begin{pmatrix}2\\0\\0\end{pmatrix}$ ist Ursache für $\begin{pmatrix}4\\0\\1\end{pmatrix}$
- $\begin{pmatrix}1\\1\\2\end{pmatrix}$ ist Ursache für $\begin{pmatrix}1\\2\\3\end{pmatrix}$
- $\begin{pmatrix}2\\1\\3\end{pmatrix}$ ist keine Ursache für $\begin{pmatrix}1\\3\\4\end{pmatrix}$

Beispiel: Kausaler Zusammenhang von Ereignissen



Klassenraumübung: Logische Zeit

Stellen Sie zum verteilen Logging-System folgende Überlegungen an:

a) Wo könnte logische Zeit zum Einsatz kommen? Begründen Sie in jedem Fall Ihre Antwort,
   - warum Sie logische Zeit einsetzen oder
   - warum Sie logische Zeit nicht einsetzen.

b) Welchen Mehrwert ergäbe die logischen Zeit im Projekt?

c) Welche logische Zeit (mit Lamport-Zeitstempel oder Vektor-Zeitstempel) ist sinnvoll, bezüglich des Mehrwerts vs. Aufwand?

Zusammenfassung

- Zeitbestimmung und Messung von Zeitdauern ist unverzichtbar zur Koordination von Aktivitäten.
- Zeitwerte verschiedener Uhren laufen auseinander, auch wenn diese synchronisiert waren (Uhrasymmetrie). Zwei Algorithmen, Cristian und Berkeley, sind zur Synchronisierung möglich.
- Lamport sagt, dass es ausreichend ist, wenn sich alle Maschinen über dieselbe Zeit einig sind. Eine Übereinstimmung mit der Zeit ausserhalb des Systems ist nicht notwendig.
- Die Happened-Before-Relation besagt, dass eine Nachricht nicht empfangen werden kann, bevor sie gesendet wurde.
- Beim Lamport-Zeitstempel wird einer Nachricht die Uhrzeit des sendenden Prozesses mitgegeben. Der empfangende Prozess richtete seine Uhrzeit nach dem Zeitstempel + 1 (mindestens).
- Die Uhrzeit muss immer vorwärts laufen.

Literatur

- Distributed Systems (3rd Edition), Maarten van Steen, Andrew S. Tanenbaum, Verleger: Maarten van Steen (ehemals Pearson Education Inc.), 2017.

Fragen?