TECHNISCHE UNIVERSITÄT CHEMNITZ

Verteilte Betriebssysteme
Wintersemester

Verteilte Betriebssysteme

7. Kapitel

Logische Uhren

Matthias Werner

Professur Betriebssysteme

7.1 Motiviation

- Erkenntnis aus letztem Kapitel: Uhrensynchronisation ist sehr aufwendig, häufig störanfällig, und begrenzt einsatzfähig

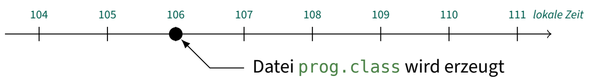
Friedemann Mattern

„The challenge consists in defining an abstract notion of time suitable for distributed systems which, on the one hand, is easily realizable without using physical clocks but, on the other hand, has enough interesting properties to justify the name 'time'.“

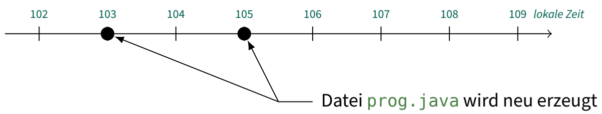
- In Rechnersystemen reicht es meist aus, die Ordnung von Ereignissen zu beschreiben
- Uhr wird zum Ereigniszähler
- Verwandt mit Ordnungen beim Broadcast ➡ Kapitel 4

Beispielproblem: Verteiltes `make`

- Computer, auf dem der Übersetzer läuft



- Computer, auf dem der Editor läuft



7.2 Lamport-Uhren

- Anforderung an eine logische Uhr (**Uhrenbedingung**) (Lamport, 1978)
  - Für alle Ereignisse a, b soll gelten:

(Schwache) Uhrenbedingung

$$a \prec b \Rightarrow C(a) < C(b)$$

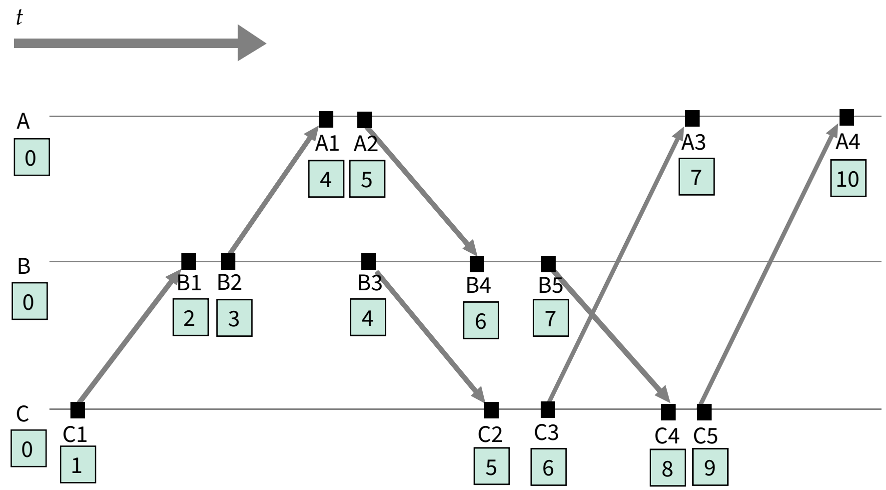
  - D.h., die Uhr erhält die **kausale** Ordnung der Ereignisse

- **Achtung:** Die Implikation gilt nur in eine Richtung!
  - Es gilt lediglich: $C(a) < C(b) \Rightarrow (a \prec b) \vee (a \parallel b)$
- Aus der Uhrenbedingung folgt: $C(a) = C(b) \Rightarrow a \parallel b$
- Wie kann eine logische Uhr realisiert werden, welche die Uhrenbedingung erfüllt?

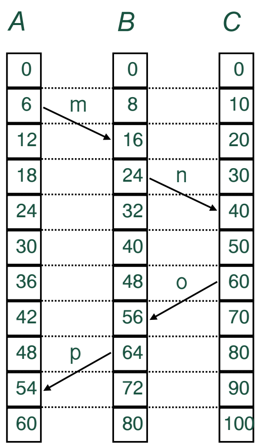
Lamports Uhren – Realisierung

- Jeder Prozess P_i verfügt über eine lokale logische Uhr L_i, deren Wert beim Auftreten der folgenden Ereignisse angepasst wird
  - Lokales Ereignis beim Prozess P_i
    - Der Wert der lokalen Uhr L_i wird um eins erhöht
    - Das Ereignis bekommt den neuen Wert als Zeitstempel
  - P_i sendet eine Nachricht m
    - Der Wert der lokalen Uhr L_i wird um eins erhöht
    - Das Sendeereignis bekommt den neuen Wert als Zeitstempel
    - m trägt als Zeitstempel t_m den Zeitstempel ihres Sendeereignisses
  - P_i empfängt eine Nachricht
    - Der Wert der lokalen Uhr wird angepasst $L_i' = \max(L_i, t_m) + 1$
    - Das Empfangsereignis erhält den neuen Wert als Zeitstempel

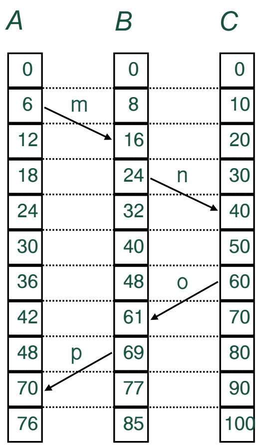
Lamports Uhren – Beispiel



Lamports Uhren – Synchronisation



ohne Synchronisation



mit Synchronisation

7.3 Hybride Uhren

- Lamport-Uhren erfüllen die Uhrenbedingung!
- Die logischen Zeitstempel $L(e)$ definieren daher eine **partielle Ordnung** auf der Menge der Ereignisse
- Häufig ist man an einer kausal korrekten aber „nutzbaren" Zeit interessiert
  - Die Aussage „*Der Fehler ist zur Zeit 42 aufgetreten*" ist von nur begrenzten Nutzen
- **Lösung**: **Hybride** oder physisch-logische[1] Uhren $H(e)$

13:42	5

  - Besteht aus physischen Zeitstempel t_P (dominant) und logischen Zeitstempeln t_L, $H(e) = (t_P, t_L)$
  - Regeln sichern, dass Kausalität eingehalten wird

[1] [Kul+14]

Hybride Uhren – Regeln

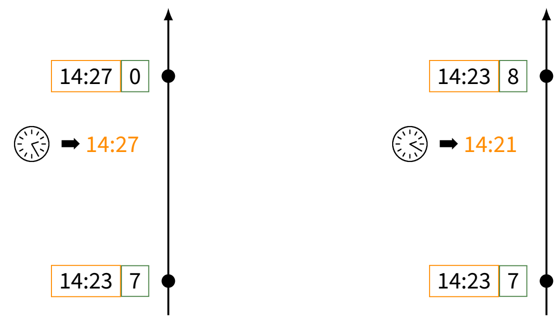
- **Annahme**: Jeder Knoten besitzt eine ungefähr synchrone Uhr
- Wie bisher werden Zeitstempel bei Nachrichten verschickt
- Wenn ein neuer Zeitstempel (t'_P, t'_L) gebildet werden soll, wird die lokale Zeit t_C gelesen
  - Neues lokales Ereignis:

$$(t'_P, t'_L) = \begin{cases} (t_C, 0) & \text{wenn } t_C > t_P \\ (t_P, t_L + 1) & \text{sonst} \end{cases}$$

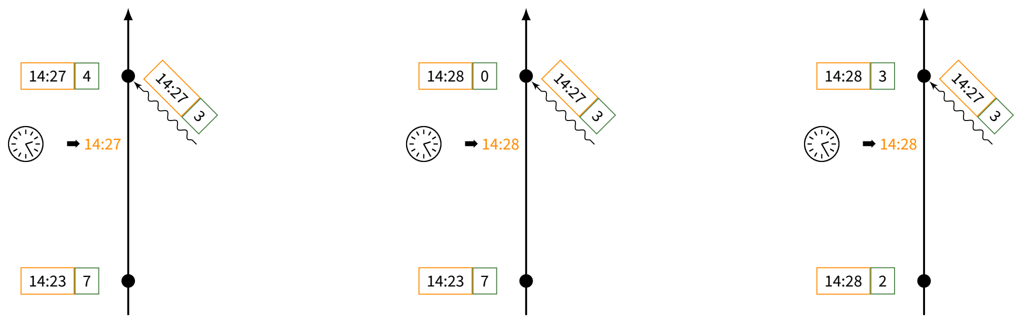
  - Empfangen einer Nachricht mit Zeitstempel $(t_{m,P}, t_{m,L})$:

$$(t'_P, t'_L) = \begin{cases} (t_C, 0) & \text{wenn } t_C > \max(t_P, t_{m,P}) \\ (t_{m,P}, t_{m,L} + 1) & \text{wenn } (t_{m,P} > t_P) \wedge (t_{m,P} \geq t_C) \\ (t_P, t_L + 1) & \text{wenn } (t_P > t_{m,P}) \wedge (t_P \geq t_C) \\ (t_P, \max(t_L, t_{m,L}) + 1) & \text{sonst} \end{cases}$$

Beispiele Hybrid-Uhren



Beispiele Hybrid-Uhren (Forts.)



Fehler

- Verteilte Systeme werden häufig nie komplett offline genommen ➡ restliche Komponenten laufen weiter
- Problematisch im Fall eines Synchronisationsfehlers bei Hybrid-Uhren:
  - **Annahme**: statt auf `19/Nov/2015:14:23:56 | 5` wird eine lokale Uhr auf `19/Nov/2051:14:23:56 | 5` gestellt
  - Mit der ersten Nachricht propagiert dieser Fehler und infiziert schnell alle beteiligten Rechner
  - **Ergebnis:** Bis in das Jahr 2051 degeneriert die Hybrid-Uhr zur reinen Lamport-Uhr
- Ohne Komplettabschaltung des verteilten Systems ist dieser Fehler **nicht** korrigierbar

Epochen

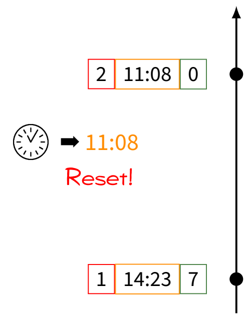
- **Idee**: Erweiterung der Hybrid-Uhren

1	14:23	5

Epoche

- „Epoche“ ist die höchstwertige Information
  - Sie wird bei der Maximumsbildung berücksichtigt
- Im Fall einer notwendigen Korrektur wird einfach die Epoche inkrementiert
  - Erlaubt einen „Reset“

Reset mit Epochen



- Kausalität weiter gewährleistet
- Für Nutzer ist erweiterte Hybrid-Uhr immernoch gut lesbar

Diskussion

- Hybrid-Uhren erfüllen genau wie die Lamport-Uhren die Uhrenbedingung ➡ Kausalität wird durch die Uhren respektiert
- **Problem:** Anhand der Zeitstempel lässt sich aber sicher nicht sagen, ob zwei Ereignisse kausal voneinander abhängen
  ➡ hierfür müsste auch die **Umkehrung** der Uhrenbedingung gelten

7.4 Vektoruhren

- Angenommen, wir hätten eine Uhr, die aus der Implikation der Uhrenbedingung eine Äquivalenz macht

Starke Uhrenbedingung

$$a \prec b \Leftrightarrow C(a) < C(b)$$

- Mit einer solchen Uhr könnten anhand der Zeitstempel von Ereignissen bestimmt werden, wie diese zueinander stehen
  - $C(a) < C(b) \Rightarrow a \prec b$
  - $C(a) > C(b) \Rightarrow b \prec a$
  - $\neg((C(a) < C(b)) \vee (C(b) < C(a))) \Rightarrow a \parallel b$
- **Hinweise:**
  - Bei allen drei Gleichungen gilt auch die Umkehrung
  - Für $\neg((C(a) < C(b)) \vee (C(b) < C(a)))$ schreibt man kurz $C(a) \parallel C(b)$

Wie lässt sich eine solche Uhr realisieren?

Vektoruhren nach MATTERN und FIDGE, 1988

- Jeder Prozess P_i hält einen Vektorzeitstempel $\vec{V_i}$ bestehend aus n Zählern
- Initial ist der Vektorzeitstempel 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 $\vec{V_i}$ mitgeschickt
- Empfängt P_i eine Nachricht mit Vektorzeitstempel $\vec{T}$, so bildet er das komponentenweise Maximum von der **neuen** Version von $\vec{V_i}$ und von $\vec{T}$
- Ansatz offensichtlich verwandt mit kausaler Auslieferungsordnung (➡ Kapitel 4)

Vektoruhren – Operationen

- Komponentenweise Maximumbildung zweier Vektoren:

$$\max(\vec{V_i}, \vec{V_j}) \stackrel{def}{=} (\max(v_{i,1}, v_{j,1}), \ldots, \max(v_{i,n}, v_{j,n}))^T$$

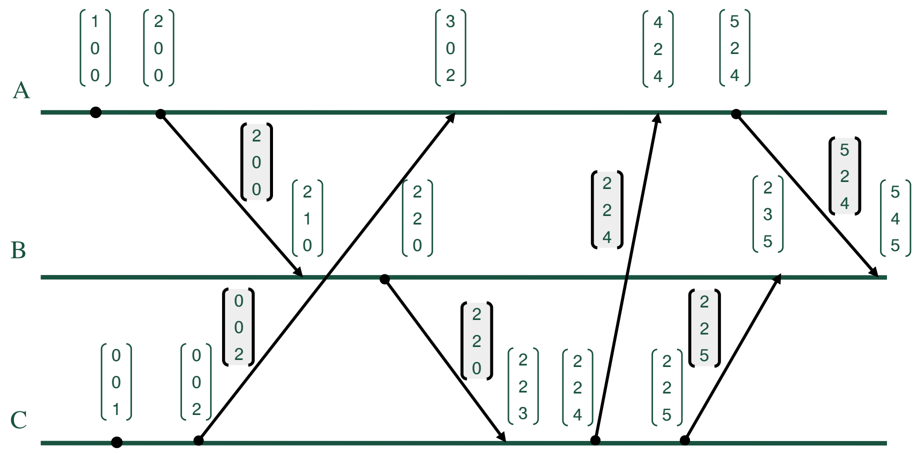
- Komponentenweiser Vergleich zweier Vektoren:

$$\vec{V_i} < \vec{V_j} \Leftrightarrow \vec{V_i} \neq \vec{V_j} \wedge \forall k, 1 \leq k \leq n, v_{i,k} \leq v_{j,k}$$

- Vektoruhren definieren lediglich eine **partielle** Ordnung

$$\begin{pmatrix}2\\2\\4\\2\\8\\6\end{pmatrix} < \begin{pmatrix}2\\3\\5\\6\\8\\9\end{pmatrix} \qquad \begin{pmatrix}1\\2\\4\\2\\5\\6\end{pmatrix} \parallel \begin{pmatrix}3\\5\\3\\1\\8\\9\end{pmatrix} \qquad \max\left(\begin{pmatrix}1\\2\\4\\2\\5\\6\end{pmatrix}, \begin{pmatrix}3\\5\\3\\1\\8\\9\end{pmatrix}\right) = \begin{pmatrix}3\\5\\4\\2\\8\\9\end{pmatrix}$$

Vektoruhren – Beispiel



Beziehung zwischen den Zeitstempeln

- $R(e)$: reale (globale) Zeit des Ereignisses e
- $L(e)$: Lamport-Zeit von e
- $V(e)$: Vektor-Zeit von e

$$\begin{array}{ccc} R(a) < R(b) & \Leftarrow & V(a) < V(b) \\ \Uparrow & \not\nearrow & \Downarrow \\ a \prec b & \Rightarrow & L(a) < L(b) \end{array}$$

7.5 Matrix-Uhren

- Es gibt noch komplexere Uhren: **Matrixuhren** (Fischer und Michael (1982))
- Jeder Prozess P_i unterhält eine $n \times n$-Matrix $\mathbb{M}_i$, die seine Sicht auf die global-verteilte Zeit des Systems repräsentiert
- Element $m_{i,i,i}$ enthält die logische Zeit des Prozesses P_i
- Zeilenvektor $\vec{m}_{i,i} = (v_{i,1}, \ldots, v_{i,n})$ enthält den Vektorzeitstempel des Prozesses P_i
- Element $m_{i,j,k}$ der Matrix $\mathbb{M}_i$ eines Prozesses P_i beschreibt, was P_j aus der Sicht von P_i über die logische Uhr $m_{k,k,k}$ von P_k weiß

$$\begin{pmatrix} 1 & 0 & 1 \\ \boxed{3 \quad 2 \quad 0} \\ 2 & 0 & 0 \end{pmatrix}$$

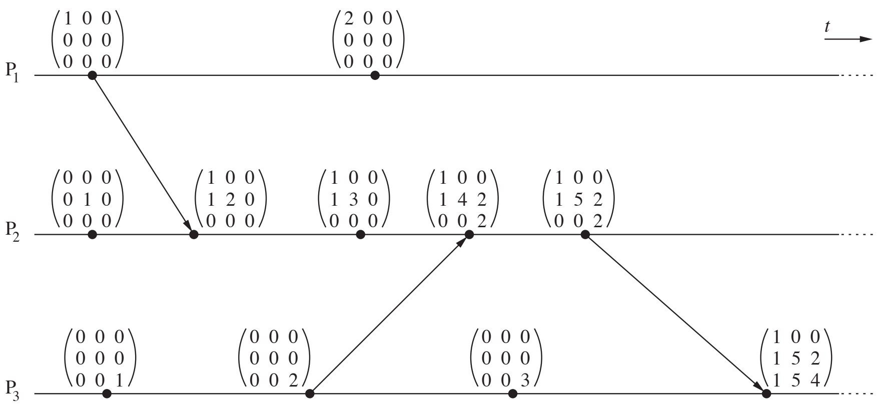
Vektorzeitstempel von P_2

logische Uhr von P_2

Regeln zur Generierung der Matrix-Zeit

- Vor der Ausführung eines Ereignisses durch P_i wird die lokale logische Uhr inkrementiert: $m'_{i,i,i} = m_{i,i,i} + 1$
- Beim Senden einer Nachricht wird der Matrixzeitstempel $\mathbb{M}T_i$ des Senders mit übertragen
- Beim Empfang eines von Prozess P_j gesendeten Matrixzeitstempels $\mathbb{M}T_j$ durch den Prozess P_i wird dessen Matrix $\mathbb{M}_i$ wie folgt aktualisiert:
  - $\forall k, 1 \leq k \leq n, m_{i,i,k} = \max(m_{i,i,k}, mt_{j,j,k})$
  - $\forall k, l, 1 \leq k, l \leq n, m_{i,k,l} = \max(m_{i,k,l}, mt_{j,k,l})$
  - $m_{i,i,i} = m_{i,i,i} + 1$

Matrix-Uhren, Beispiel



Und was hat man nun davon?

- Zunächst können wir alles, was Vektorzeit auch kann ➡ starke Uhrenbedingung
- Zusätzlich gilt:

$\min_k(m_{i,k,l}) \geq t \Rightarrow$ Prozess P_i weiß, dass jeder andere Prozess P_k weiß, dass die Zeit von P_i mindestens bis t fortgeschritten ist

- Nützlich bei
  - konsistenter Fehlererkennung im System
  - Eliminierung überflüssiger Information, z.B.:
    - In replizierten Datenbeständen (z.B. Lynch und Sarin 1987)
    - Vorhalten des Rückgabewertes bei RPC für den Fehlerfall (vgl. Kapitel 3)

Literatur

[Mat89] Mattern , F. : *Verteilte Basisalgorithmen*. Springer, 1989 , Kapitel 5: Virtuelle Zeit in verteilten Systemen

[Lam78] Lamport , L. : „Time, clocks, and the ordering of events in a distributed system“. *Commun. ACM*, 21(7)1978, 558–565

[Kul+14] Kulkarni , S. S. , M. Demirbas, D. Madappa, B. Avva und M. Leone: „Logical Physical Clocks“, in *Principles of Distributed Systems*, edited by M. K. Aguilera, L. Querzoni and M. Shapiro, 17–32, Springer International Publishing