

## Willkommen bei Verteilte Systeme!

Von Datenbanken über Webdienste bis zu p2p und Sensornetzen.



Heute: Koordination – Reihenfolge, Uhren, Konfliktvermeidung, Wahlen.

Arne Babenhausen/Heide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Wiederholung

## Wiederholung 3 - Architekturstile

- Layered, Object-, Resource-, Event-based
- n-Schichten
- Overlay Networks

Arne Babenhausen/Heide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Wiederholung

## Ablauf heute

### Koordination

- Das Problem mit der Reihenfolge
  - Timestamps als Lösung?
    - Uhren im Computer
    - Synchronisation von Uhren
    - Alternative Uhren
    - total geordneter Multicast
    - kausal geordneter Multicast
- Exklusiver Zugriff
- Wahlalgorithmen

Arne Babenhausen/Heide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Wiederholung

## Reihenfolge 1

```
balance = 1000
balance = balance + 100
balance = balance * 1.05
return balance
```

- 1155.0
- Sicht aus Rechenzentrum KA.
  - Message des Kunden trifft zuerst ein.
  - Message des Bankers danach.
  - Kontostand: 1'155 €.

Arne Babenhausen/Heide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Wiederholung

## Lernziele Uhren + Synchronisation

- Sie kennen die Funktionsweise von Uhren in Computern
- Sie kennen Methoden, um Uhren zu synchronisieren

Arne Babenhausen/Heide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Wiederholung

## Clock Skew vs. Clock Drift

Wir vergleichen 2 Uhren:

- Clock Skew: Unterschied der Werte der Uhren
- Clock Drift: Unterschied zwischen der Frequenzen der Uhren
- Clock Skew  $\neq 0 \Rightarrow$  Uhren sind nicht synchronisiert sind
- Clock Drift  $\neq 0 \Rightarrow$  Clock Skew wird sich verändern
- Clock Skew vs. zu UTC ca. 31 Sekunden pro Jahr
  - Ursache: Unterschiede in der Frequenz des Quartz (auch bei baugleichen Uhren)
  - Externe Einflüsse wie Temperatur
  - $\Rightarrow$  Wir müssen Synchronisieren!

Arne Babenhausen/Heide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Wiederholung

## Wiederholung 1

- Verteilte Systeme sind eine Sammlung autonomer Knoten, die als einzelnes kohärentes System erscheinen.
- Ziele verteilter Systeme:
  - Teilen von Ressourcen
  - Verteilungstransparenz
  - Offenheit
  - Skalierbarkeit

Arne Babenhausen/Heide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Wiederholung

## Wiederholung 4 - Kommunikation

- Kommunikation kann transient/persistent und (a)synchronus sein
- RPC ist ein entfernter Methodenaufwurf
- Messaging erlaubt entkoppelte Kommunikation
- ZeroMQs Primitive erlauben: Request-Reply, Publish-Subscribe, Pipeline
- Link Stress und Stretch als Metriken für Overlay Netzwerke

Arne Babenhausen/Heide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Wiederholung

## Ziele heute

- Sie verstehen, wie Operationen in einem verteilten System geordnet werden können
- Sie kennen die Funktionsweise von Uhren in Computern
- Sie kennen Methoden, um Uhren zu synchronisieren
- Sie kennen Alternativen zu synchronisierten Uhren
- Sie kennen Implementierungen für wechselseitigen Ausschluss (mutual exclusion) in verteilten Systemen
- Sie kennen grundlegende Wahlalgorithmen, um Knoten spezielle Rollen zuzuweisen

Arne Babenhausen/Heide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Wiederholung

## Reihenfolge 2

```
balance = 1000
balance = balance * 1.05
balance = balance + 100
return balance
```

- 1150.0
- Sicht aus Rechenzentrum FFM.
  - Message des Bankers trifft zuerst ein.
  - Message des Kunden danach.
  - Kontostand: 1'150 €.

Arne Babenhausen/Heide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Wiederholung

## Uhren

Computer verwenden 2 Arten von Uhren:

- time-of-day clocks
  - Gibt aktuelles Datum und Uhrzeit zurück.
  - Beispiel: `Java System.currentTimeMillis()`
- monotonic clocks
  - Geben eine Zahl zurück die monoton steigt.
  - Beispiel: `Java System.nanoTime()`

Arne Babenhausen/Heide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Wiederholung

## Zeit-Synchronisation

Es existieren verschiedene Algorithmen:

- Cristians Algorithmus
- NTP
- Berkeley

Arne Babenhausen/Heide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Wiederholung

## Wiederholung 2

- Dimensionen der Skalierung: Größe, Geographie, Administration
- Techniken der Skalierung: Verstecken von Latenz, Partitionierung, Replikation
- Fallacies of distributed systems
- Arten: Cluster, Grid, Cloud, Ubiquitous, Mobile Computing, verteilte Informationssysteme, Sensornetze

Arne Babenhausen/Heide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Wiederholung

## Organisation: Projekte

- Konkrete Ziele entwerfen
- Atomic Planning Poker
- Zielnoten nach Zeitschätzung<sup>1</sup>
- Ziele anpassen

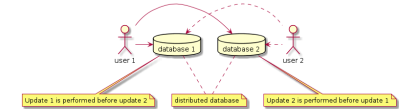
<sup>1</sup>Ich schaue, was der niedrigste Wert wäre, Rest verteilen wir dazwischen. Zu hohes sind Ausreißer und sollten die anderen nicht beeinflussen!

Arne Babenhausen/Heide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Wiederholung

## Problem mit Reihenfolge



- Repliziertes Bankkonto, gespeichert in 2 Rechenzentren (KA, FFM).
- Kontostand zu Beginn: 1'000 €.
- Kunde in KA möchte 100 € einzahlen.
- Banker in FFM möchte 5% Zinsen auf das Konto buchen.

Arne Babenhausen/Heide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Wiederholung

## Das Problem

- Je nach Reihenfolge der Messages, resultiert ein anderer Kontostand.
- Solche Inkonsistenzen sollten vermieden werden.
- Wie lassen sich die Operationen ordnen?

Arne Babenhausen/Heide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Wiederholung

## Hardware Uhren

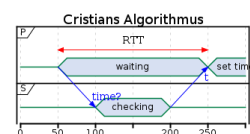
- Bestehen aus einem Quartz und 2 Registern.
- Der Quartz oszilliert in einer bestimmten Frequenz.
- Das counter-Register hält eine Zahl, die bei jeder Oszillation dekrementiert wird.
- Erreicht der Zähler 0, wird ein Interrupt abgesetzt.
  - Danach wird das counter-Register auf den Wert des holding-Registers gesetzt.
- Jeder Interrupt stellt einen tick dar.
- Die Software Uhr wird pro tick um 1 erhöht.

Arne Babenhausen/Heide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Wiederholung

## Cristians Algorithmus



- P fragt Zeit von S an.
- S liest die Zeit und antwortet.
- P setzt seine Uhr auf  $t + \frac{RTT}{2}$

Arne Babenhausen/Heide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Wiederholung

Arne Babenhauerheide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wählen Abschluss

Synchronisation

## PAUSE

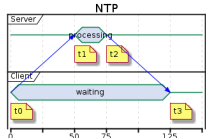
--- PAUSE ---

Arne Babenhauerheide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wählen Abschluss

Synchronisation

## NTP Berechnung



- Client startet Anfrage zu  $t_0$ .
- Server empfängt Anfrage zu  $t_1$  und sendet Antwort zu  $t_2$ .
- Client empfängt Antwort zu  $t_3$ .
- $t_1$  und  $t_2$  jeweils in Client Zeit  $t_c$  und  $t_s$  in Server Zeit

Arne Babenhauerheide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wählen Abschluss

Synchronisation

## NTP

Neben dem offset wird noch das delay berechnet:  
 $delay = (t_3 - t_0) - (t_2 - t_1)$   
 Es werden 8 offset-delay Paare ermittelt und das Paar mit dem geringsten delay verwendet.  
 NTP erreicht Genauigkeiten von 1-50 ms.

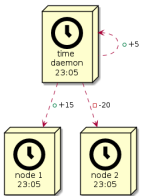
Arne Babenhauerheide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wählen Abschluss

Synchronisation

## Berkeley Algorithmus - Schritt 3

Zeitserver berechnet Durchschnitt der Uhrzeiten und sendet offsets an Maschinen.



Arne Babenhauerheide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wählen Abschluss

Synchronisation

## Wall Time Timestamps - Fazit

- Uhren sind immer mit einem Fehler versehen.
- Timestamps sollten eigentlich die Form  $[T_{lower} T_{upper}]$  haben.
  - Spanner verwendet True Time Service mit solchen Intervallen.
  - Und schafft es auf 6ms Genauigkeit zu kommen.
  - Transaktionen werden verzögert, damit  $T_{upper}$  auf jeden Fall verstrichen ist.

Arne Babenhauerheide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wählen Abschluss

Lampport Uhren

## Lernziele Logische Uhren + Multicast

- Alternativen zu wall time clocks.
- Was ist eine partielle und was eine totale Ordnung?
- Wie funktionieren Lamport clocks?
- Wie implementiert man total geordneten Multicast mit Lamport clocks.
- Wie funktionieren Vector clocks?
- Wie implementiert man kausal geordneten Multicast mit vector clocks.

Arne Babenhauerheide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wählen Abschluss

Lampport Uhren

Arne Babenhauerheide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wählen Abschluss

Synchronisation

## Übung Cristians Algorithmus

- Laufzeit Nachricht  $P \rightarrow S$  und  $S \rightarrow P$  jeweils 100 ms.
- S benötigt 10 ms für die Bearbeitung der Anfrage.
- $t = 500$  ms.
- Welche Zeit wird bei P eingestellt?

Arne Babenhauerheide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wählen Abschluss

Synchronisation

## NTP graduelle Anpassung

- Was passiert bei  $offset < 0$ ?
- Uhren sollten nicht rückwärts laufen!
- Bsp: jeder tick erhöht Software Uhr um 10 ms.
  - Idee: Verringerung des Inkrements, um Uhr schrittweise anzugleichen.
- Wird auch verwendet, um die Uhr vorwärts anzupassen.
- Graduelle Anpassung wird bei offsets  $> 128$  ms nicht verwendet.

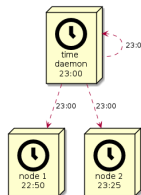
Arne Babenhauerheide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wählen Abschluss

Synchronisation

## Berkeley Algorithmus - Schritt 1

Zeitserver sendet periodisch eigene Zeit an alle Maschinen im Netzwerk.



Arne Babenhauerheide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wählen Abschluss

Synchronisation

## Berkeley Algorithmus - Übung

3 Rechner und ein Server im System.  
 Gebt die Nachrichten des Berkeley Algorithmus an. Welche Zeit wird im System eingestellt?

Server	11:00		
Alice	10:55		
Bob	11:15		
Carol	11:10		
Nr	von	an	Inhalt
1	Server	A,B,C	11:00
2	Server	Server	0
...			
9			

Arne Babenhauerheide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wählen Abschluss

Synchronisation

Zusammenfassung Uhren + Synchronisation

## Zusammenfassung Uhren + Synchronisation

- wall time vs. monotonic clocks
- Uhren leiden unter skew und drift.
  - > müssen synchronisiert werden.
- externe Synchronisation
  - Cristians Algorithmus
  - NTP
- interne Synchronisation
  - Berkeley

Arne Babenhauerheide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wählen Abschluss

Lampport Uhren

## Lampport Uhren

- Die exakte Uhrzeit interessiert uns nicht
- Reihenfolge von Ereignissen
- => Zeit-Ordnung
- Timestamps sollen Kausalität berücksichtigen:
  - Wenn a kausal vor b passiert ist, dann  $timestamp(a) < timestamp(b)$ .
  - Tür wird geöffnet bevor man eintritt

Arne Babenhauerheide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wählen Abschluss

Lampport Uhren

Arne Babenhauerheide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wählen Abschluss

Synchronisation

## NTP (Network Time Protocol)



- Bestandteile des Systems werden in Strata unterteilt.
- Uhren befinden sich in Stratum 0.
- Ein Server in Stratum n kontaktiert Server in Stratum n - 1 zur Synchronisation.
- Oft werden mehrere Server angefragt und die Ergebnisse statistisch behandelt (Mittel, Ausreißer).

Arne Babenhauerheide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wählen Abschluss

Synchronisation

## NTP Übung

- Client stellt Anfrage bei  $t_0 = 100$  ms.
- Server empfängt Anfrage bei  $t_1 = 50$  ms und benötigt 10 ms zur Bearbeitung.
- Berechne das offset bei:
  - $t_c \rightarrow s = t_s \rightarrow c = 5$  ms
  - $t_c \rightarrow s = 5$  ms und  $t_s \rightarrow c = 10$  ms
- Auf welche Zeit wird die Uhr des Client jeweils gestellt?

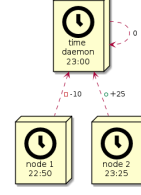
Arne Babenhauerheide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wählen Abschluss

Synchronisation

## Berkeley Algorithmus - Schritt 2

Maschinen antworten mit ihrem offset.



Arne Babenhauerheide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wählen Abschluss

Synchronisation

## Berkeley Algorithmus - Beobachtungen

- Hierbei handelt es sich um interne Synchronisation.
  - System wird nicht mit externer Uhr (UTC) synchronisiert.
- Solange das System intern einen einheitlichen Zeitbegriff verwendet, können Operationen geordnet werden.

Arne Babenhauerheide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wählen Abschluss

Synchronisation

Zusammenfassung Uhren + Synchronisation

## PAUSE

--- PAUSE ---

Arne Babenhauerheide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wählen Abschluss

Lampport Uhren

## Die Happens-Before Relation

- Logische Relation zwischen 2 Ereignissen.
- Notation:  $a \rightarrow b$ , a ist vor b passiert.
- Regeln:
  - Innerhalb eines Prozesses  $a \rightarrow b$ , if  $time(a) < time(b)$ .
  - Wenn P1 eine Nachricht m an P2 sendet:  $send(m) \rightarrow receive(m)$ .
  - Wenn  $a \rightarrow b$  und  $b \rightarrow c$ , dann auch  $a \rightarrow c$  (Transitivität)
- Partielle Ordnung von Ereignissen.

Arne Babenhauerheide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wählen Abschluss

Lampport Uhren

Arne Babenhauerheide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Lampot Uhren

## Partielle Ordnung

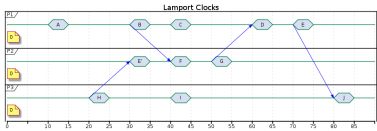
- Eine **totale Ordnung** erlaubt 2 beliebige Elemente miteinander zu vergleichen.
  - Für jedes Elementpaar kann die Aussage getroffen werden welches der Elemente größer ist.
  - Beispiel: natürliche Zahlen.
- Eine **Partielle Ordnung** kann nur einige vergleichen
- $\Rightarrow$  Wir können nicht für alle Ereignispaare die Reihenfolge bestimmen.

Arne Babenhauerheide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Lampot Uhren

## Lampot Clocks - Schritt 1

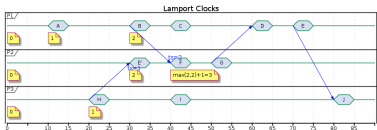


Arne Babenhauerheide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Lampot Uhren

## Lampot Clocks - Schritt 4

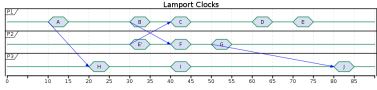


Arne Babenhauerheide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Lampot Uhren

## Lampot Clocks - Übung



- Besteht ein kausaler Zusammenhang zwischen:
  - A, J
  - H, G
  - C, F
- Berechne die timestamps.

Arne Babenhauerheide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Vektor Uhren

## Vector Uhren - Verwaltung

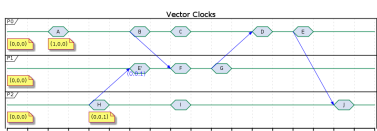
- Vor Ausführung eines lokalen Ereignisses führt  $P_i$  die Anweisung  $V_i[i] += 1$  aus.
- Wenn  $P_i$  eine Nachricht sendet, wird der timestamp der Nachricht auf  $V_i$  gesetzt, nachdem  $V_i[i] += 1$  ausgeführt wurde.
- Beim Empfang einer Nachricht in  $P_j$ :
  - $V_j[i] += 1$
  - $V_j[j] = \max(V_m[j], V_i[j]), \text{ for } j \neq i$

Arne Babenhauerheide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Vektor Uhren

## Vector Uhren - Beispiel 3



Arne Babenhauerheide  
Verteilte Systeme 2: Koordination

Arne Babenhauerheide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Lampot Uhren

## Umsetzung Lampot Clocks

Jeder Prozess  $P_i$  erstellt einen lokalen Zähler  $C_i$  und wendet folgende Regeln an:

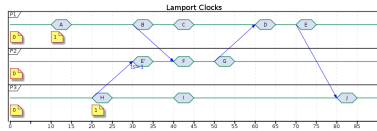
- Für 2 **suksessive Ereignisse**, die in  $P_i$  stattfinden, wird  $C_i$  um 1 erhöht.
- Wenn eine Nachricht von  $P_i$  **gesendet** wird, erhält sie den timestamp  $ts(m) = C_i$ .
- Wenn eine Nachricht von  $P_j$  **empfangen** wird, setz  $C_j$  auf  $\max(C_j, ts(m)) + 1$

Arne Babenhauerheide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Lampot Uhren

## Lampot Clocks - Schritt 2

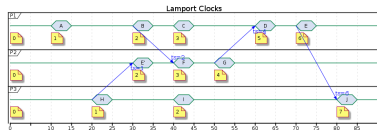


Arne Babenhauerheide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Lampot Uhren

## Lampot Clocks - Ende



Arne Babenhauerheide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Vektor Uhren

## Vektor Uhren

Lampot Uhren:

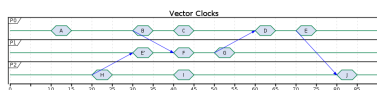
- $A \rightarrow B \implies ts(A) < ts(B)$  **aber**
  - $ts(A) < ts(B) \implies \{A \rightarrow B\}$  OR  $\{A, B \text{ gleichzeitig}\}$
- Gibt es ein Verfahren, das  $ts(A) < ts(B) \implies A \rightarrow B$  ermöglicht?

Arne Babenhauerheide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Vektor Uhren

## Vector Uhren - Beispiel 1



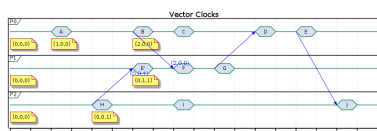
Selbe Ausgangssituation wie bei Lampot Clocks.

Arne Babenhauerheide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Vektor Uhren

## Vector Uhren - Beispiel 4



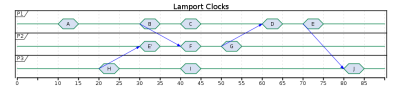
Arne Babenhauerheide  
Verteilte Systeme 2: Koordination

Arne Babenhauerheide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Lampot Uhren

## Beispiel Lampot Clocks

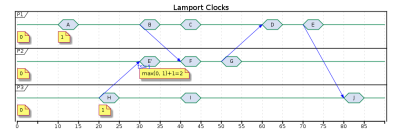


Arne Babenhauerheide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Lampot Uhren

## Lampot Clocks - Schritt 3



Arne Babenhauerheide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Lampot Uhren

## Lampot Clocks

- Ein Paar von gleichzeitigen (concurrent) Ereignissen hat keinen kausalen Pfad.
- Lamport timestamps müssen bei gleichzeitigen Ereignissen weder geordnet noch ungleich sein.
- $A \rightarrow B \implies ts(A) < ts(B)$  **aber**
  - $ts(A) < ts(B) \implies \{A \rightarrow B\}$  OR  $\{A, B \text{ gleichzeitig}\}$

Arne Babenhauerheide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Vektor Uhren

## Vector Uhren - Umsetzung

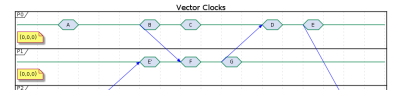
- Jeder Prozess verwaltet einen Vektor von Integer Uhren.
- Bei N Prozesse hat jeder Vektor N Elemente.
- Ein Prozess i verwaltet einen Vektor  $V_i[0..(N-1)]$
- $V_i[i]$  ist die lokale Uhr des Prozesses i.
- Falls  $V_i[j]=k$ , dann weiß i, dass k Ereignisse in  $P_j$  stattgefunden haben.
- In jedem Knoten  $O(N) \rightarrow$  Wir hätten gerne  $O(\log(N))$  (haben wir aber nicht).

Arne Babenhauerheide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Vektor Uhren

## Vector Uhren - Beispiel 2

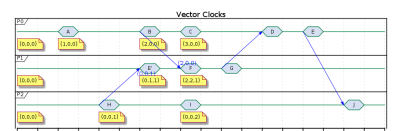


Arne Babenhauerheide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Vektor Uhren

## Vector Uhren - Beispiel 5

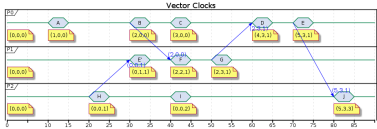


Arne Babenhauerheide  
Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Vektor Uhren

## Vector Uhren - Beispiel 6

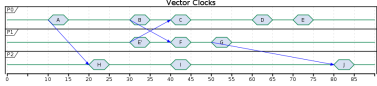


Arne Babenhauerheide Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Vektor Uhren

## Vector Uhren - Übung



- Berechne die timestamps.
- Hängt J vielleicht kausal von A ab?
- Finden C und F gleichzeitig statt?

Arne Babenhauerheide Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Kausal geordneter Multicast

## Kausal geordneter Multicast

- Vollständig geordneter Multicast stellt sicher, dass alle Nachrichten in gleicher Reihenfolge bearbeitet werden.
- Kausal geordnet bedeutet, dass Nachrichten, die sich gegenseitig beeinflussen könnten in gleicher Reihenfolge von allen Prozessen empfangen werden.

Arne Babenhauerheide Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Kausal geordneter Multicast

## Kausal geordneter Multicast - Anpassungen

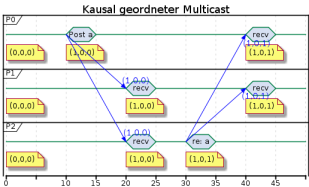
- Vector Clocks werden nur bei Empfang oder Senden einer Nachricht angepasst.
  - Beim Senden in  $P_i$ :  $V_i[i] += 1$
  - Beim Empfang von  $m$  in  $P_j$ : für alle  $k$ :  $V_j[k] = \max(V_i[k], V_{m_j}[k])$
- Eine Nachricht  $m$  (von  $P_k$  an  $P_j$ ) wird erst von der Anwendung prozessiert wenn:
  - $V_{m_j}[k] = V_j[k] + 1$ ,  $m$  ist die nächste Nachricht, die  $P_{-j}$  von  $P_{-k}$  erwartet hat.
  - $V_{m_j}[x] \leq V_j[x]$  für alle  $x \neq k$ ,  $P_j$  hat alle Nachrichten gesehen, die  $P_k$  gesehen hat als  $m$  gesendet wurde.

Arne Babenhauerheide Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Kausal geordneter Multicast

## Kausal geordneter Multicast - Beispiel 3



Was passiert bei verzögerter Zustellung?

Arne Babenhauerheide Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Zusammenfassung Logische Uhren + Multicast

## PAUSE

--- PAUSE ---

Arne Babenhauerheide Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Vektor Uhren

## Vector Uhren - kausale Abhängigkeit

Ereignis B hängt vielleicht kausal von A ab, wenn  $ts(A) < ts(B)$ .

$ts(A) < ts(B)$ :

- für alle  $i$ :  $ts(A)[i] \leq ts(B)[i]$  und
- es existiert mindestens ein Index  $k$  für den:  $ts(A)[k] < ts(B)[k]$  gilt.

In diesem Fall kann man sagen:

- A geht B kausal voraus.
- B hängt vielleicht kausal von A ab, da es Informationen von A geben könnte, die in B propagiert werden.

Arne Babenhauerheide Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Vektor Uhren

## Vector Uhren - Abschluss

Lampot Clocks:

- $A \rightarrow B \implies ts(A) < ts(B)$  aber
- $ts(A) < ts(B) \implies \{A \rightarrow B\} \text{ OR } \{A, B \text{ gleichzeitig}\}$

Vector Clocks:

- $ts(A) < ts(B) \implies A \rightarrow B$
- $(\neg(ts(A) < ts(B))) \wedge (\neg(ts(B) < ts(A))) \implies A, B \text{ gleichzeitig}$

Arne Babenhauerheide Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Kausal geordneter Multicast

## Kausal geordneter Multicast mit Vector Clocks

Mit wenigen Änderungen können Vector Clocks genutzt werden, um kausal geordnete Nachrichten sicherzustellen.

Folgendes Beispiel:

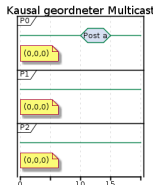
- Ein Bulletin Board Service.
- Benutzer treten Gruppen bei.
- Posts werden an alle Gruppenmitglieder gemulticastet.
- Könnte vollständig geordneten Multicast verwenden.
  - Wird aber nicht benötigt.

Arne Babenhauerheide Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Kausal geordneter Multicast

## Kausal geordneter Multicast - Beispiel 1

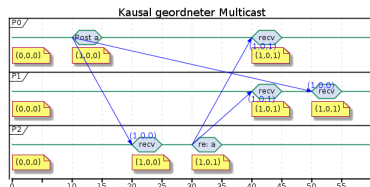


Arne Babenhauerheide Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Kausal geordneter Multicast

## Kausal geordneter Multicast - Beispiel 4



Arne Babenhauerheide Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Lernziele Gegenseitiger Ausschluss + Wahlen

## Lernziele Gegenseitiger Ausschluss + Wahlen

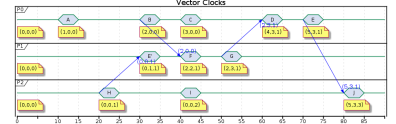
- Welche grundlegenden Algorithmen gibt es?

Arne Babenhauerheide Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Vektor Uhren

## Vector Uhren - Gleichzeitigkeit



Paar	$ts(1)$	$ts(2)$	$ts(1) < ts(2)$	$ts(2) < ts(1)$	Folgerung
A, E	(1,0,0)	(5,3,1)	Ja	Nein	A kausal vor E
H, C	(0,0,1)	(3,0,0)	Nein	Nein	H und C gleichzeitig

Arne Babenhauerheide Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Vektor Uhren

## PAUSE

--- PAUSE ---

Arne Babenhauerheide Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Kausal geordneter Multicast

## Bulletin Board - Anzeige

Betreff  
Mach  
Microkernels  
Hurd; was: Microkernels  
RPC Performance  
Re: Mach

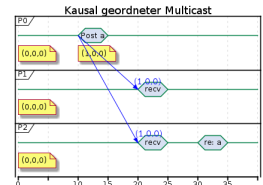
- Bei vollständiger Ordnung ist diese Liste bei jedem Benutzer in der gleichen Reihenfolge.
- Kausal geordneter Multicast erfordert nur, dass Reaktionen (Re: Mach) nach dem Post (Mach) angezeigt werden.
- Für die angezeigten Posts sind verschiedene Reihenfolgen möglich.

Arne Babenhauerheide Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Kausal geordneter Multicast

## Kausal geordneter Multicast - Beispiel 2



Arne Babenhauerheide Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Zusammenfassung Logische Uhren + Multicast

## Zusammenfassung Logische Uhren + Multicast

- happens before Relation bestimmt eine partielle Ordnung.
- Lampot Clocks: Counter pro Prozess
  - Timestamps bilden totale Ordnung.
  - Vergleich von timestamps gibt keine Aussage zu Kausalität.
- Total geordneter Multicast kann mit Lampot Clocks implementiert werden.
- Vector Clocks: Vector of Counter pro Prozess
  - Vergleich von timestamps gibt Aussage zu Kausalität.
  - Kausal geordneter Multicast kann mit vector clocks implementiert werden.

Arne Babenhauerheide Verteilte Systeme 2: Koordination

Einstieg Problem mit Reihenfolge Logische Uhren + Multicast Gegenseitiger Ausschluss + Wahlen Abschluss

Gegenseitiger Ausschluss

## Gegenseitiger Ausschluss

Das Problem: Einige Prozesse in einem verteilten System möchten exklusiven Zugriff auf eine Ressource.

2 Vorgehensweisen:

- Berechtigungs-basiert: Prozesse benötigen Berechtigung anderer Prozesse um auf Ressource zuzugreifen.
- Token-basiert: Einzigartiges Token wird zwischen Prozessen weitergereicht. Wer das Token hält, hat Zugriff auf die Ressource.<sup>2</sup>

<sup>2</sup>Token-Basiert bis hin zu Funktionsargumenten: <https://fosdem.org/2022/schedule/event/spritalyglblins/>

Arne Babenhauerheide Verteilte Systeme 2: Koordination

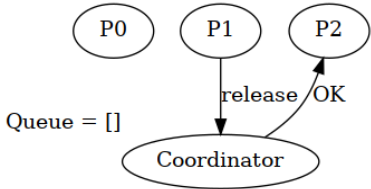
Einstieg	Problem mit Reihenfolge	Logische Uhren + Multicast	Gegenseitiger Ausschluss + Wahlen	Abschluss
0	0000	0	0000000000000000	0
0000	0000	0	0000000000000000	0
00	00	00	0000000000000000	00
Gegenseitiger Ausschluss				

## Zentralisierter Algorithmus

- Simuliert Vorgehen innerhalb einer CPU.
- Ein Prozess wird als Koordinator konfiguriert.
- Prozesse, die auf die Ressource zugreifen möchten, fragen dies beim Koordinator an.

Arne Babenhausenheide				
Verteilte Systeme 2: Koordination				
Einstieg	Problem mit Reihenfolge	Logische Uhren + Multicast	Gegenseitiger Ausschluss + Wahlen	Abschluss
0	0000	0	0000000000000000	0
0000	0	0	0000000000000000	0
00	00	00	0000000000000000	00
Gegenseitiger Ausschluss				

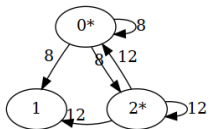
## Zentralisierter Algorithmus - Ressource wird frei



Sobald die Ressource wieder frei wird, gibt der Koordinator die Ressource an den ersten Prozess in der Queue.

Arne Babenhausenheide				
Verteilte Systeme 2: Koordination				
Einstieg	Problem mit Reihenfolge	Logische Uhren + Multicast	Gegenseitiger Ausschluss + Wahlen	Abschluss
0	0000	0	0000000000000000	0
0000	0	0	0000000000000000	0
00	00	00	0000000000000000	00
Gegenseitiger Ausschluss				

## Verteilter Algorithmus - Gleichzeitiger Zugriff



- P0 sendet Anfragen mit timestamp 8.
- P2 sendet Anfragen mit timestamp 12.

Arne Babenhausenheide				
Verteilte Systeme 2: Koordination				
Einstieg	Problem mit Reihenfolge	Logische Uhren + Multicast	Gegenseitiger Ausschluss + Wahlen	Abschluss
0	0000	0	0000000000000000	0
0000	0	0	0000000000000000	0
00	00	00	0000000000000000	00
Gegenseitiger Ausschluss				

## Verteilter Algorithmus - Bewertung

- Was passiert wenn ein Knoten ausfällt?
  - Können wir den Algorithmus anpassen?
- Wieviele Nachrichten werden benötigt?

Arne Babenhausenheide				
Verteilte Systeme 2: Koordination				
Einstieg	Problem mit Reihenfolge	Logische Uhren + Multicast	Gegenseitiger Ausschluss + Wahlen	Abschluss
0	0000	0	0000000000000000	0
0000	0	0	0000000000000000	0
00	00	00	0000000000000000	00
Gegenseitiger Ausschluss				

## Vergleich

Algorithmus	Nachrichten pro Ein/Austritt
Zentralisiert	3
Verteilt	2(N-1)
Token Ring	1, ..., ∞

Arne Babenhausenheide  
Verteilte Systeme 2: Koordination  
Literatur

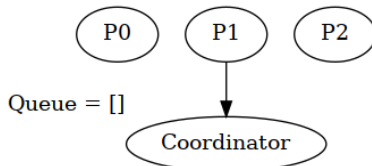
## Verweise I

Bilder:

Arne Babenhausenheide  
Verteilte Systeme 2: Koordination

Einstieg	Problem mit Reihenfolge	Logische Uhren + Multicast	Gegenseitiger Ausschluss + Wahlen	Abschluss
0	0000	0	0000000000000000	0
0000	0	0	0000000000000000	0
00	00	00	0000000000000000	00
Gegenseitiger Ausschluss				

## Zentralisierter Algorithmus - Happy Path



Falls die Ressource verfügbar ist, erhält der anfragende Prozess die Berechtigung.

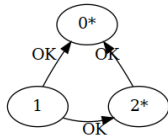
Arne Babenhausenheide				
Verteilte Systeme 2: Koordination				
Einstieg	Problem mit Reihenfolge	Logische Uhren + Multicast	Gegenseitiger Ausschluss + Wahlen	Abschluss
0	0000	0	0000000000000000	0
0000	0	0	0000000000000000	0
00	00	00	0000000000000000	00
Gegenseitiger Ausschluss				

## Zentralisierter Algorithmus - Bewertung

- Wieviele Nachrichten werden ausgetauscht?
- Was passiert, wenn der Koordinator ausfällt?
- Was passiert bei vielen Anfragen?
  - 3 Messages pro lock
  - System funktioniert nicht mehr
  - Koordinator ist Flaschenhals

Arne Babenhausenheide				
Verteilte Systeme 2: Koordination				
Einstieg	Problem mit Reihenfolge	Logische Uhren + Multicast	Gegenseitiger Ausschluss + Wahlen	Abschluss
0	0000	0	0000000000000000	0
0000	0	0	0000000000000000	0
00	00	00	0000000000000000	00
Gegenseitiger Ausschluss				

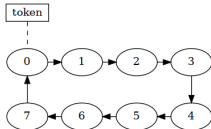
## Verteilter Algorithmus - Gleichzeitiger Zugriff 2



- P1 sendet OK an beide Prozesse.
- P0 und P2 vergleichen timestamps.
  - P2 sendet OK.
  - P0 stellt P2s Anfrage in einer Queue ein.

Arne Babenhausenheide				
Verteilte Systeme 2: Koordination				
Einstieg	Problem mit Reihenfolge	Logische Uhren + Multicast	Gegenseitiger Ausschluss + Wahlen	Abschluss
0	0000	0	0000000000000000	0
0000	0	0	0000000000000000	0
00	00	00	0000000000000000	00
Gegenseitiger Ausschluss				

## Token Ring Algorithmus



- Prozesse werden in ringförmigen Overlay Netzwerk angeordnet.
- Erster Prozess erhält Token.
- Token stellt Berechtigung dar die Ressource zu verwenden.
- Wird Ressource nicht benötigt, wird Token weitergeleitet.

Arne Babenhausenheide				
Verteilte Systeme 2: Koordination				
Einstieg	Problem mit Reihenfolge	Logische Uhren + Multicast	Gegenseitiger Ausschluss + Wahlen	Abschluss
0	0000	0	0000000000000000	0
0000	0	0	0000000000000000	0
00	00	00	0000000000000000	00
Zusammenfassung Gegenseitiger Ausschluss + Wahlen				

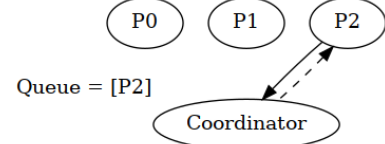
## Zusammenfassung Gegenseitiger Ausschluss + Wahlen

- Mutex mit Koordinator
- Mutex verteilt
- Mutex Token Ring

Arne Babenhausenheide  
Verteilte Systeme 2: Koordination

Einstieg	Problem mit Reihenfolge	Logische Uhren + Multicast	Gegenseitiger Ausschluss + Wahlen	Abschluss
0	0000	0	0000000000000000	0
0000	0	0	0000000000000000	0
00	00	00	0000000000000000	00
Gegenseitiger Ausschluss				

## Zentralisierter Algorithmus - Ressource belegt



- Koordinator hat Übersicht, ob Ressource momentan frei ist.
- Hier wird die Antwort an den anfragenden Prozess verzögert, bis die Ressource wieder frei wird.

Arne Babenhausenheide				
Verteilte Systeme 2: Koordination				
Einstieg	Problem mit Reihenfolge	Logische Uhren + Multicast	Gegenseitiger Ausschluss + Wahlen	Abschluss
0	0000	0	0000000000000000	0
0000	0	0	0000000000000000	0
00	00	00	0000000000000000	00
Gegenseitiger Ausschluss				

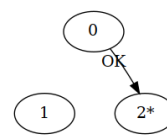
## Verteilter Algorithmus

- Wenn ein Prozess, die Ressource benötigt sendet er eine Nachricht an alle Prozesse (auch sich selbst).
- Bei Erhalt so einer Nachricht:
  - Prozess hält Ressource nicht und möchte sie nicht: sendet OK.
  - Prozess hält Ressource: antwortet nicht.
  - Prozess möchte Ressource: vergleiche timestamp der Nachricht mit timestamp der eigenen Nachricht. Der niedrigere timestamp gewinnt.
- Prozess wartet Antworten aller Prozesse ab. Sobald er sämtliche OKs erhalten hat, verwendet er die Ressource.

Voraussetzung: Totale Ordnung der Nachrichten.

Arne Babenhausenheide				
Verteilte Systeme 2: Koordination				
Einstieg	Problem mit Reihenfolge	Logische Uhren + Multicast	Gegenseitiger Ausschluss + Wahlen	Abschluss
0	0000	0	0000000000000000	0
0000	0	0	0000000000000000	0
00	00	00	0000000000000000	00
Gegenseitiger Ausschluss				

## Verteilter Algorithmus - Gleichzeitiger Zugriff 3



- P0 benötigt die Ressource nicht mehr.
- P0 sendet OK an erste Anfrage in Queue.
- P2 erhält Zugriff.

Arne Babenhausenheide				
Verteilte Systeme 2: Koordination				
Einstieg	Problem mit Reihenfolge	Logische Uhren + Multicast	Gegenseitiger Ausschluss + Wahlen	Abschluss
0	0000	0	0000000000000000	0
0000	0	0	0000000000000000	0
00	00	00	0000000000000000	00
Gegenseitiger Ausschluss				

## Token Ring Algorithmus - Bewertung

- Was passiert wenn ein Knoten ausfällt?
- Wieviele Nachrichten werden benötigt?

Arne Babenhausenheide				
Verteilte Systeme 2: Koordination				
Einstieg	Problem mit Reihenfolge	Logische Uhren + Multicast	Gegenseitiger Ausschluss + Wahlen	Abschluss
0	0000	0	0000000000000000	0
0000	0	0	0000000000000000	0
00	00	00	0000000000000000	00
Abschluss				

## Für koordinierte Projekte!

Arne Babenhausenheide  
Verteilte Systeme 2: Koordination