

Aufgabenblatt 10

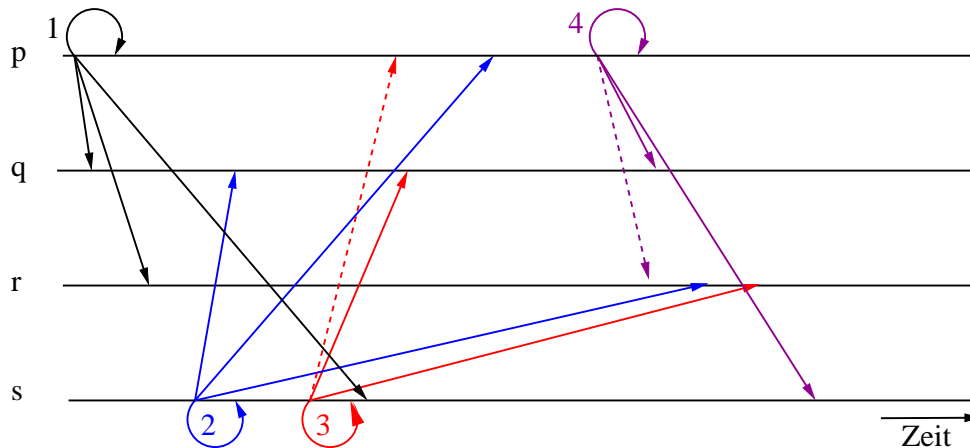
(Zu bearbeiten bis 15.01.)

Vorlesung Verteilte Systeme Wintersemester 2025/26

Aufgabe 1: Reihenfolgegarantien beim Multicast

In der Vorlesung wurden verschiedene Varianten für Reihenfolge-Garantien beim Multicast vorgestellt (s. Kap. 7.3).

- a) In der Abbildung sind 4 Prozesse (p, q, r, s) gezeigt und vier Multicast-Nachrichten (1, 2, 3, 4).



Nachricht 4 ist im Beispiel kausal abhängig von Nachricht 3.

Wann muß die Auslieferung der Nachrichten 3 und 4 stattfinden, so dass eine FIFO und eine kausale Sortierung für alle Nachrichten gewährleistet wird? Hinweis: Sie müssen nur die gestrichelten Pfeile richtig zeichnen. Ist dann auch eine vollständige Sortierung gewährleistet?

- b) Um die Multicast-Sortierung zu verdeutlichen, betrachten wir die Anwendung „elektronisches schwarzes Brett“, bei der die Benutzer Nachrichten auf einem schwarzen Brett hinterlassen. Jeder Benutzer führt einen Applikationsprozess für das schwarze Brett aus. Es gibt mehrere Diskussionsthemen, zu denen jeweils eine eigene Prozeßgruppe gehört. Der Prozess eines Benutzers ist Mitglied in der Gruppe für das Thema, an dem dieser interessiert ist, so dass der Benutzer nur Nachrichten zu dem jeweiligen Thema erhält. Wenn ein Benutzer eine Nachricht auf einem schwarzen Brett anbringt, multicastet die Applikation diese an die entsprechende Gruppe.

Ein Benutzer X könnte dann folgende Anzeige aus dem Programm für das schwarze Brett erhalten:

Schwarzes Brett zum Thema: Kommunikation in verteilten Systemen

Nachricht	von	Titel der Nachricht
27	A. Bauer	UDP & TCP Kommunikation
28	C. Fritz	RPC Prinzip
29	G. Meier	Re: UDP & TCP Kommunikation
30	K. Wim	RPC 1
31	A. Bauer	Re: RPC Prinzip
32	K. Wim	RPC 2

Beachten Sie, dass die Nachrichten, deren Themen mit *Re:* anfangen, nach den Nachrichten erscheinen, auf die sie sich beziehen.

Welche sortierte Multicast-Verfahren werden in der Applikation gebraucht und wo genau? Wenn der Multicast vollständig sortiert wäre, was kann man dann über die Nummerierung in der linken Spalte sagen?

Aufgabe 2: Vollständig sortierter Multicast

Um vollständig sortierten Multicast mit der Hilfe eines Sequenzers zu implementieren, ist ein Ansatz, die Nachricht zuerst an den Sequenzer weiterzugeben, der ihr eine eindeutige Nummer zuweist und sie anschließend per Multicast weitergibt. Nennen Sie einen alternativen Ansatz und vergleichen Sie die zwei Lösungen.

Aufgabe 3: Vollständig sortierter Multicast / Lamport Zeitstempel

In einem Szenario zur Implementierung einer Bank wird ein vollständig sortierter Multicast benötigt, d.h. eine Multicast-Operation, wobei alle Nachrichten in der selben Reihenfolge an alle Empfänger ausgeliefert werden. Dabei werden Lamportzeitstempel verwendet, um den Multicast auf verteilter Weise zu implementieren. Wenn ein Prozess eine Nachricht empfängt, wird sie in eine lokale Warteschlange gestellt, die nach Zeitstempeln organisiert ist.

Der Empfänger sendet per Multicast eine Bestätigung an die anderen Prozesse. Wenn wir den Algorithmus von Lamport folgen, um die lokalen Uhren anzupassen, sind die Zeitstempel der empfangenen Nachricht kleiner als die Zeitstempel der Bestätigung. Nach einer kurzen Zeit stellt sich heraus, dass die lokalen Warteschlangen, in denen sich die Nachrichten und Benachrichtigungen befinden, bei allen Prozessen gleich sind, da davon ausgegangen werden kann, dass die Auslieferung in der gleichen Reihenfolge geschieht wie die, in der sie gesendet wurden. Ein Prozess kann dann nur eine Nachricht ausliefern, wenn sie sich am Anfang der Warteschlange befindet und von allen anderen Prozessen bestätigt wurde.

Ist es hier streng erforderlich, dass jede Nachricht bestätigt wird, um ein vollständig sortiertes Multicasting mit Lamport-Zeitstempeln zu erzielen?

Pflichtaufgabe 4: Konsistenzmodelle Abgabe über moodle!

- In der Vorlesung (Kap.8.2) wurde eine Reihe von Konsistenzmodellen vorgestellt. Warum werden diese unterschiedlichen Formen von Konsistenz eingeführt? Erklären Sie in eigenen Worten, was der wichtigste Grund ist, abgeschwächte Konsistenzmodelle in Betracht zu ziehen.
- Gegeben seien die zwei nachfolgenden Berechnungen in einem verteilten System. Begründen Sie jeweils, ob das gezeigte Speicherverhalten der sequentiellen Konsistenz bzw. der kausalen Konsistenz genügt.

P1: W(x)a	W(x)b
P2: R(x)a W(x)c	R(y)a
P3: R(x)a	W(y)a R(x)c R(x)b

P1: W(x)a	W(x)b
P2: R(y)a	R(x)a
P3: R(x)b	W(y)a

Bezeichnungen:

- $W(x)a$: in die Variable x wird der Wert a geschrieben.
- $R(x)b$: Variable x wird gelesen, das Ergebnis ist b .

Pflichtaufgabe 5: Sequentielle Konsistenz Abgabe über moodle!

Ist der Speicher, der der folgenden Ausführung von zwei Prozessen zu Grunde liegt, sequentiell konsistent? Es sei vorausgesetzt, dass alle Variablen zunächst auf Null gesetzt sind.

$P_1 : W(a)1; R(b)2; W(a)3$

$P_2 : R(a)1; R(a)3; W(b)2$