

Aufgabenblatt 8

(Zu bearbeiten bis 18.12.)

Vorlesung Verteilte Systeme Wintersemester 2025/26

Aufgabe 1: Uhrensynchronisation

- a) Ein Client versucht, eine Synchronisierung mit einem Zeit Server durchzuführen. Er hat dazu drei Anfragen an den Server ausgeführt und in der nachfolgenden Tabelle jeweils die lokale Zeit des Absendens der Anfrage, die vom Server zurückgesendete Zeit und die gemessene Round-Trip-Zeit aufgezeichnet.

Welche dieser Zeiten sollte er verwenden, um seine Uhr zu setzen? Auf welche Zeit sollte er sie setzen? Schätzen Sie die Genauigkeit, wenn bekannt ist, dass die Zeit zwischen dem Senden einer Nachricht und dem Empfangen der Antwort in dem betreffenden System mindestens $8ms$ beträgt.

Absendezeit [hh:mm:ss]	Round-Trip-Zeit[ms]	Zeit des Servers [hh:mm:ss]
11:51:20.043	22	11:51:20.564
11:51:21.821	25	11:51:22.340
11:51:24.715	20	11:51:25.232

- b) Wenn die minimale Übertragungszeit von $8ms$ gegeben sei und die Uhr eines Datei-Servers auf $\pm 1ms$ genau synchronisiert werden muss, wie hoch darf die Round-Trip-Zeit dann höchstens sein?

Aufgabe 2: Happened-Before-Relation, Lamport- und Vektorzeit

Drei Prozesse A, B und C erzeugen jeweils die Ereignisse $a_1, a_2, \dots, b_1, b_2, \dots$ und c_1, c_2, \dots

- a) Nehmen Sie an, daß die Ereignisse $a_1, a_2, a_3, a_4, b_1, b_2, c_1, c_2$ aufgetreten sind und folgende Lamport-Zeiten haben:

Ereignis	a_1	a_2	a_3	a_4	b_1	b_2	c_1	c_2
Lamport-Zeit	1	2	3	4	1	2	1	5

Welche Aussagen lassen sich über die *happened-before* Relation zwischen

- a_3 und c_2 ,
- a_1 und b_1 ,
- b_2 und c_1

machen? Welche Aussagen können Sie jeweils über die Reihenfolge der beiden Ereignisse in der Realzeit machen?

- b) Für die Ereignisse aus a) gelten nun folgende Vektorzeiten:

Ereignis	a_1	a_2	a_3	a_4	b_1	b_2	c_1	c_2
Vektor-Zeit	(1,0,0)	(2,1,0)	(3,1,0)	(4,1,0)	(0,1,0)	(0,2,1)	(0,0,1)	(4,1,2)

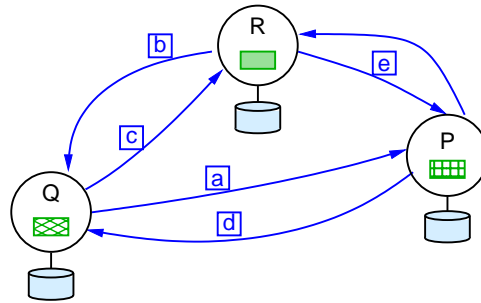
Welche Aussagen lassen sich nun über die *happened-before* Relation der Ereignisse aus a) treffen? Welche Aussagen sind über die Reihenfolge in der Realzeit möglich?

- c) Wie unterscheiden sich Lamport- und Vektorzeit in ihrer Beziehung zur Lamport'schen *happened before* Relation?

Pflichtaufgabe 3: Schnappschuß-Algorithmus nach Chandy/Lamport Abgabe über moodle!

In der Vorlesung (Kap. 6.4) haben Sie gelernt, wie nach Chandy und Lamport ein konsistenter Zustand in einem verteilten System ermittelt wird.

Gegeben sei ein verteiltes System mit drei Prozessen P , Q , und R , die über zwei unidirektionale Kanäle verbunden sind. Das System befindet sich in dem in der Abbildung gezeigten Ausgangszustand. Die an den Kommunikationsverbindungen angegebenen Nachrichten befinden sich in Übertragung, d.h. sind gesendet, aber noch nicht empfangen.



- Erklären Sie die Arbeitsweise des Algorithmus in diesem Fall und geben Sie den von ihm zurückgegebenen globalen Zustand an (Hinweis: dieser Zustand ist nicht eindeutig bestimmt!). Nehmen Sie an, dass Prozess P den Schnappschuß-Algorithmus initiiert.
- Zeichnen Sie ein Ablaufdiagramm der Prozesse und geben Sie die Schnitte für die folgenden globalen Zustände an:
 - den dargestellten Ausgangszustand
 - den vom Schnappschuß-Algorithmus bestimmten Zustand

Aufgabe 4: Schnappschuß-Algorithmus nach Chandy/Lamport

Zwei Prozesse P und Q sind unter Verwendung von zwei Kanälen in einem Ring verbunden, und sie schicken ständig eine Nachricht m zwischen sich hin und her. Es gibt zu jedem Zeitpunkt nur eine Kopie von m im System. Der Zustand jedes Prozesses besteht aus der Anzahl, wie oft er m bereits empfangen hat. P fängt an m zu senden. Zu einem bestimmten Zeitpunkt hat P die Nachricht und sein Zustand ist 101. Unmittelbar nach dem Senden von m initiiert P den Schnappschuß-Algorithmus.

Führen Sie den Algorithmus in diesem Fall aus und geben Sie den von ihm zurückgegebenen globalen Zustand an.

Aufgabe 5: Bully-Algorithmus

In der Vorlesung wurde der Bully-Algorithmus vorgestellt (s. Kap. 7.1). Diskutieren Sie dazu folgende Fragestellungen:

- Angenommen zwei Prozesse erkennen gleichzeitig, dass der Koordinator ausgefallen ist, und halten beide eine Wahl unter Verwendung des Bully-Algorithmus ab. Was passiert?
- Im Bully-Algorithmus startet ein wiederhergestellter Prozess eine Wahl und wird zum neuen Koordinator, wenn er eine höhere ID hat als der aktuelle Inhaber. Ist dies ein notwendiges Funktionsmerkmal des Algorithmus?
- Schlagen Sie vor, wie der Bully-Algorithmus angepasst werden kann, so dass er mit einer temporären Unterbrechung des Netzwerks (langsame Kommunikation) und ebenso langsamen Prozessen zurecht kommt.

Pflichtaufgabe 6: Programmierung: Ricart/Agrawala Algorithmus **Abgabe über moodle!**

In dieser Aufgabe sollen Sie mit Hilfe des Verteilte-Systeme-Simulators den Ricart/Agrawala Algorithmus (siehe Abschnitt 8.2 der Vorlesungsfolien) implementieren.

Bearbeiten Sie dazu die Aufgabe „[Ricart Agrawala algorithm](#)¹“ im Wiki (Zugriff nur aus dem Uni-VPN!).

¹<https://git.bs.informatik.uni-siegen.de/dsbox/exercises/wiki/5-ricart-agrawala>