

**Verteilte Algorithmen und Anwendungen
Wintersemester 2019/2020**

Prof. Dr. Markus Esch
Moritz Fey, M. Sc.

Übung 3

Aufgabe 1: Wechselseitiger Ausschluss

In dieser Aufgabe soll das Problem des wechselseitigen Ausschlusses an folgendem Beispiel untersucht werden. Wir betrachten ein System aus n unabhängigen Knoten, die zufällig miteinander verbunden sind. Jeder Knoten P_i besitzt ein Konto, welches beim Starten mit einem zufälligen positiven Geldbetrag zwischen 0 und 100.000 initialisiert wird. Die Anzahl der Knoten im System ist konstant und jedem Knoten bekannt.

Jeder Knoten P_i führt folgende Schritte durch:

1. Warte eine zufällige Zeit zwischen 0 und 3 Sekunden
2. Wähle einen zufälligen Knoten im Netzwerk P_j
3. Erlange ein Lock für die Konten von P_i und P_j
4. Wähle einen zufälligen Prozentsatz p zwischen 0 und 100
5. Sende den eigenen Kontostand B_i sowie p an P_j
6. Frage den Kontostand B_j von P_j ab
7. P_i führt folgende Operation durch
 - o Wenn $B_i \geq B_j$: Erhöhe den eigenen Kontostand um p -Prozent von B_j
 - o Wenn $B_i < B_j$: Verringere den eigenen Kontostand um p -Prozent
8. P_j führt folgende Operation durch
 - o Wenn $B_i \geq B_j$: Erhöhe den eigenen Kontostand um p -Prozent von B_j
 - o Wenn $B_i < B_j$: Verringere den eigenen Kontostand um p -Prozent
 - o Sende ein Acknowledgment an P_i , zur Bestätigung, dass die Operation durchgeführt wurde
9. Warte auf das Acknowledgment von P_j
10. Gebe die Locks frei
11. Fahre mit 1 fort

Beachten Sie bei der Implementierung folgendes:

- In Schritt 2 soll ein zufälliger Knoten nicht nur unter den direkten Nachbarn ausgewählt werden, sondern im gesamten Netzwerk. Wählen Sie dazu eine zufällige Knoten-ID zwischen 1 und n .
- Realisieren Sie den wechselseitigen Ausschluss entweder durch den Lamport-Algorithmus oder den Ricart & Agrawala-Algorithmus.
 - o Ein Request wird per Flooding im Netzwerk verteilt.
- Beachten Sie in Punkt 7 und 8 des Algorithmus, dass die Operationen von beiden Knoten unabhängig ausgeführt werden sollen. D.h. nur die in Schritt 5 und 6 ausgetauschten Informationen werden zur Durchführung der Operationen verwendet.

Implementieren Sie den Algorithmus auf Basis Ihrer Lösung für Aufgabenblatt 1.

Aufgabe 2: Deadlocks Detection

In Aufgabe 1 können Deadlocks auftreten. Implementieren Sie den Goldman Edge Chasing-Algorithmus, um Deadlocks zu erkennen und aufzulösen. Ein Knoten initiiert den Algorithmus, wenn er nach einer vorgegebenen Zeit nicht alle angeforderten Locks erhalten hat. Definieren Sie eine sinnvolle Wartezeit.

Aufgabe 3: Globaler Zustand

Erweitern Sie Ihre Lösung um die Möglichkeit die Geldmenge im System festzustellen. Implementieren Sie dazu den Chandy-Lamport-Algorithmus. Wählen Sie zu Beginn einen Koordinator (Hier können Sie Ihre Lösung von Aufgabenblatt 2 wiederverwenden). Der Koordinator ermittelt periodisch die Geldmenge im System und gibt diese aus.

Ändert sich die Geldmenge, soll der Koordinator eine Warnung ausgeben. Überlegen Sie, worauf diese Warnung hindeutet.

Beachten Sie, dass der Koordinator die Marker-Nachricht nicht direkt an alle Knoten senden kann, sondern in dem Netzwerk flutet. Nachdem ein Knoten seinen lokalen Snapshot abgeschlossen hat, sendet er diesen an den Koordinator. Der Koordinator ermittelt dann den globalen Zustand.

Anmerkung: Geben Sie bei allen Aufgaben die Schritte der Algorithmen so aus, dass diese gut nachvollzogen werden können.