

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

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

Übung 2

Aufgabe 1: Distributed Consensus - Implementierung

Ein bekanntes Problem in Bereich verteilter Algorithmen ist das des Distributed Consensus. Dabei geht es darum, in einem verteilten System ohne zentrale Kontrolle Einigung unter allen beteiligten Prozessen zu erzielen. Dieses Problem soll in einer einfachen Version untersucht werden. Dazu betrachten wir folgende Analogie:

Eine Gruppe von n Philosophen will sich auf einen gemeinsamen Zeitpunkt für eine wissenschaftliche Konferenz einigen. Diese Einigung soll jedoch ohne zentrale Kontrolle und ohne Gruppenkommunikation erfolgen. Wir gehen davon aus, dass eine feste und diskrete Menge von Zeitpunkten 1 bis m zur Auswahl steht und dass potentiell jeder zu jedem der zur Verfügung stehenden Zeiten verfügbar ist. Zu Beginn wählt jeder Philosoph P_k einen bevorzugten Zeitpunkt t_k . Zur Abstimmung kommunizieren die Philosophen immer nur paarweise miteinander. Die Philosophen sind außerdem nicht alle direkt miteinander bekannt, sondern in einem zufälligen zusammenhängenden Graphen verbunden.

Um den Einigungsprozess initiieren und beobachten zu können, wird ein Koordinator benötigt. Dieser soll zu Beginn unter den Philosophen gewählt werden. Dazu initiiert eine zufällige Teilmenge der Philosophen eine verteilte Wahl. Der Gewinner der Wahl soll durch den Wahlprozess alle anderen Philosophen kennen lernen.

Nach Abschluss der Wahl startet der Koordinator den Einigungsprozess, indem er s zufällig gewählten Philosophen eine Startnachricht sendet; s soll ein konfigurierbarer Parameter sein.

Jeder Philosoph P_a der die Startnachricht erhält, verfährt wie folgt: Er wählt p (p soll ein konfigurierbarer Parameter sein) zufällige Nachbarn aus und stimmt mit einem nach dem anderen einen Zeitpunkt ab. Zur Abstimmung mit Nachbar P_b tauschen P_a und P_b ihre bevorzugten Zeitpunkte t_a und t_b aus und einigen sich anschließend auf den Mittelwert der beiden Zeitpunkte. Da wir mit diskreten Zeitpunkten arbeiten, wird bei nicht ganzzahligen Lösungen der Mittelwertbildung immer aufgerundet. Nach erfolgter Abstimmung mit P_a führt P_b die Abstimmung auf gleiche Weise fort. Ein Philosoph darf gleichzeitig an mehreren Abstimmungen teilnehmen, ein wechselseitiger Ausschluss ist also nicht erforderlich.

Es gibt einen konfigurierbaren Parameter A_{max} , dieser gibt an, wie viele Abstimmungsrunden jeder Philosoph mitmacht. Nach Erreichen der maximalen Abstimmungsrunden ignoriert ein Philosoph die weiteren Abstimmungsnachrichten.

Der Koordinator prüft periodisch mit Hilfe der Double Counting-Methode, ob die Abstimmung bereits beendet ist. Nachdem die Abstimmung beendet ist, sammelt der Koordinator das Ergebnis ein und propagiert es im Netzwerk. Das Einsammeln des Ergebnisses soll mithilfe des Echo-Algorithmus erfolgen. Das Ergebnis ist entweder der gefundene gemeinsame Zeitpunkt oder die Feststellung, dass kein gemeinsamer Zeitpunkt gefunden wurde.

Modellieren und Implementieren Sie das oben beschriebene Problem auf Basis Ihrer Lösung für Aufgabenblatt 1. Um den Ablauf nachvollziehen zu können, soll jeder Prozess alle Schritte der Abstimmung ausgeben.

Aufgabe 2: Distributed Consensus - Evaluation

Führen Sie Experimente mit unterschiedlichen Werten für n , m , s , p und A_{max} durch. Jeder Prozess sollte dabei jeden Schritt des Abstimmungsprozesses protokollieren.

Stellen Sie Ihre Ergebnisse in einem oder mehreren Graphen dar, welche die zeitliche Entwicklung des Abstimmungsprozesses für unterschiedlichen Parameter zeigen. Mitteln Sie dazu die Ergebnisse von 10 unabhängigen Testläufen.

Ziehen Sie ein Fazit, wie Erfolg des Abstimmungsprozesses, Nachrichtenkomplexität und Dauer der Abstimmung von den Parametern abhängen.