



Verteilte Systeme, Übungsblatt 2, Sommer 2024

Aufgabe 1

Wie in der Vorlesung besprochen, soll der Simulator `sim4da-v2` die Grundlage für dieses zweite Übungsblatt bilden. Klonen Sie das Repository <https://github.com/oxoo2a/sim4da-v2.git> und machen Sie sich mit der grundlegenden Struktur der Anwendung vertraut. Teil dieses Repositories ist auch ein Pre-Release **Beta-1**, das ein Überjar mit allen verwendeten Packages beinhaltet. Wenn Sie dieses Jar-File in Ihrer eigenen Anwendung als verwendetes Package registrieren, sollte eine Simulation – hoffentlich – nicht all zu schwierig sein.

Aufgabe 2 - Fireworks

Entwerfen Sie auf der Basis des Simulators aus Aufgabe 1 eine einfache verteilte Anwendung, die aus einer konfigurierbaren Anzahl von n Knoten besteht. Diese Knoten sollen als Aktoren modelliert werden, die nur im aktiven Zustand Nachrichten versenden und im passiven Fall nur durch eine eintreffende Nachricht wieder aktiviert werden können. Zu Beginn wartet jeder dieser Aktoren eine Zufallszeit und sendet anschließend eine Nachricht **Firework** an eine zufällige Teilmenge anderer Knoten. Anschließend wechselt der Aktor in den passiven Zustand. Eine eintreffende Nachricht **Firework** kann einen passiven Aktor wieder in den aktiven Zustand versetzen. Dabei entscheidet der Aktor mit einer Wahrscheinlichkeit p , dass er erneut eine Nachricht **Firework** an eine beliebige Teilmenge anderer Aktoren sendet oder dies sein lässt. Nach jeder Entscheidung wird diese Wahrscheinlichkeit p verkleinert (z.B. $p=p/2$ oder $p=p/10$) und der Aktor wechselt erneut in den passiven Zustand.

Aufgabe 3 – No more Fireworks?

Erweitern Sie Ihre Lösung aus Aufgabe 2 um eine nachrichtenbasierte Terminierungserkennung. Gefordert wird also ein zusätzlicher Knoten **Observer**, der feststellt, dass alle Aktoren passiv sind und keine weiteren Nachrichten **Firework** unterwegs sind.

Abgabe

Abzugeben ist der Link zu einem Repository, das die Implementierungen aus Aufgabe 2 und 3 sowie einen kurzen Ergebnisberichts (PDF, ca. 2 Seiten) beinhaltet. Wie in der Vorlesung besprochen, dürfen Sie auch den Code des Simulatorkerns selbst verändern, wenn Sie entsprechende Verbesserungsvorschläge haben oder Ihnen ein Fehler aufgefallen ist. Bitte gehen Sie auf diese Änderungen im Simulatorkern im Ergebnisbericht gesondert ein. Wünschenswert ist in diesem Fall auch ein Pull-Request, damit ich diese Verbesserungen in das offizielle Repository `sim4da-v2` übernehmen kann.

Die Abgabefrist ist zu Beginn der Übung am 19. Juli 2024. In dieser Übung werden wir über die verschiedenen Lösungsansätze und eventuelle Realisierungsprobleme sprechen. Aufgrund der verspäteten Ausgabe des Übungsblatts werde ich auf Kurzvorträge zu den einzelnen Lösungen verzichten.