Universität Trier

Fachbereich IV / Informatik
Prof. Dr. Peter Sturm
D-54286 Trier
Telefon 0651/201-3313
Fax 0651/201-3842
Email sturm@uni-trier.de

Verteilte Systeme, Übungsblatt 1, Sommer 2025

Aufgabe 1 – Ein Feuerwerk an UDP-Nachrichten (Pseudo-Verteilt)

In dieser Aufgabe sollen n Prozesse in einem logischen Ring (Overlay-Netzwerk) eine Art Streichholz (Token) kreisen lassen. Ein Prozess, der das Token bekommt, zündet mit einer Wahrscheinlichkeit p eine Feuerwerksrakete (Broadcast-Nachricht an alle Prozesse im Ring) und leitet anschließend das Streichholz weiter. Jeder Prozess reduziert mit jedem Durchlauf seine Zündwahrscheinlichkeit p (z.B. p = p/2). Verwenden Sie UDP für die Kommunikation im Ring und bilden Sie alle Broadcasts auf UDP-Multicast ab. Die verteilte Anwendung soll terminieren, wenn in k aufeinanderfolgenden Runden kein einziger Prozess eine Feuerwerksrakete gezündet hat. In dieser ersten Aufgabe sollen alle n Prozesse auf dem Testrechner (localhost, 127.0.0.1) gestartet werden. Führen Sie mehrere Experimente mit wachsendem n (z.B. n = 2, 4, 8, 16, …) durch. Schreiben Sie dazu ein Shell- oder Python-Script, um den Aufbau und die Ausführung der immer größer werdenden Ringe zu automatisieren. Zusätzlich zu Ihrer Implementierung sollten Sie im Rahmen der Experimente (a) das maximale n ermitteln, das gerade noch so auf Ihrem Testrechner erfolgreich durchgeführt werden konnte sowie (b) wesentliche statistische Informationen erfassen (mindestens: Gesamtanzahl der Token-Runden; Gesamtanzahl der gesendeten Multicasts; minimale, mittlere und maximale Rundenzeit; jeweils in Abhängigkeit von n).

Aufgabe 2 – Ein Feuerwerk an UDP-Nachrichten (Verteilt)

Führen Sie die Anwendung aus Aufgabe 1 auf möglichst vielen realen Computersystemen aus. Auf jedem Computer soll nur ein Prozess aus dem logischen Ring ausgeführt werden. Bilden Sie weiterhin – nach Möglichkeit – die Broadcasts auf UDP-Multicast-Nachrichten ab. Wenn das nicht gehen sollte, können Sie alternativ n-1 Unicast-Nachrichten an die anderen Prozesse senden. Ermitteln Sie analog zu Aufgabe 1 das maximale n, das Sie erreichen (hängt vermutlich primär davon ab, zu wieviel anderen Computersystemen Sie Zugang haben; ggf. überzeugen Sie Menschen aus Ihrem Umfeld, an diesem hochinteressanten Experiment teilzunehmen) sowie ebenfalls die in Aufgabe 1 gemessenen statistischen Informationen (insbesondere minimale, mittlere und maximale Rundenzeit).

Aufgabe 3 – Ein simuliertes Feuerwerk

Besorgen Sie sich den in der Vorlesung vorgestellten Simulator https://github.com/syssoft-ds/sim4da-S25. Nutzen Sie den Test OneRingToRuleThemAll als Ausgangspunkt für eine Simulation der in Aufgabe 1 geforderten verteilten Anwendung. Bestimmen Sie auch hier analog zu den Aufgaben 1 und 2 das maximal erreichbare n (eventuelle Thread- und Speicher-Limits kann man bei einer JVM durch entsprechende Parameter erhöhen) sowie die statistischen Informationen (bei den Rundenzeiten interessieren auch hier die real gemessenen Zeiten). Vergleichen und Interpretieren Sie die Ergebnisse aus Aufgabe 3 mit den Ergebnissen aus den Aufgaben 1 und 2. Vergleichen Sie auch den Implementierungs- und den Experimentalaufwand der verschiedenen Realisierungen.

Aufgabe 4 – Konsistenz

Geht in den Implementierungen und der Simulation alles mit rechten Dingen zu oder können die beteiligten Prozesse in bestimmten Situationen eine inkonsistente Sicht auf den Programmablauf haben? Denken Sie in dieser Aufgabe über die Definition von Konsistenzkriterien für diese verteilte Anwendung nach. Ergänzen Sie anschließend Ihre Implementierung aus Aufgabe 3 um Mechanismen, die inkonsistente Situationen zumindest erkennen und melden oder idealerweise sogar vermeiden.

Abgabe

Eine erfolgreiche Abgabe dieses ersten Teils der Portfolioprüfung besteht aus: (1) dem Sourcecode zu den Aufgaben 1 bis 4, (2) einem 3-5-seitigen Bericht und (3) einem ca. 15-minütigen Vortrag in der Übung. Den Sourcecode geben Sie mittels E-Mail an

mich ab, vorzugsweise mit einem Link auf ein Github-Repository oder inklusive einer ZIP-Datei. Alternative Formen der Abgabe sollten Sie vorher mit mir absprechen. Der Bericht soll sich auf die von Ihnen gewählten Implementierungen konzentrieren, auf die Thematik „Konsistenz in einer konkreten verteilten Anwendung“ eingehen und die gemessenen Werte diskutieren. Verwenden Sie zur Visualisierung komplexer Strukturen und Abläufe geeignete UML-Diagramme. Im Vortrag stellen Sie ebenfalls Ihre Lösungen vor und diskutieren mit den Teilnehmern Ihre Ergebnisse und Erfahrungen.

Die Abgabefrist ist der 25.5.2025. Sollten bis zur Abgabefrist Fragen oder Probleme aufkommen, können Sie sich gerne über die üblichen Kommunikationswege an uns wenden.