UNIVERSITÄT SIEGEN

Fakultät IV
Betriebssysteme und verteilte Systeme
Prof. Dr. rer. nat. Roland Wismüller

Aufgabenblatt 11

(Zu bearbeiten bis 10.07.2020)

Vorlesung Verteilte Systeme
Sommersemester 2020

Aufgabe 1: Replikation

a) Erläutern Sie kurz, warum auf die Benutzung von Replikaten zurückgegriffen wird.

b) Nennen und erläutern Sie kurz die vorgestellten, unterschiedlichen Arten von Replikaten.

Aufgabe 2: Aktive Replikation

Damit die aktive Replikation funktioniert, ist es i.A. erforderlich, dass alle Operationen auf allen Repliken in derselben Reihenfolge ausgeführt werden. Ist diese Reihenfolge immer erforderlich?

Aufgabe 3: Quorum-Protokolle

Durch den Ausfall von Netzverbindungen oder den Ausfall eines Gateway-Rechners, der mehrere Netze miteinander verbindet, können funktionsfähige Teilnetze (Partitionen) entstehen, deren Knoten noch untereinander kommunizieren können. Wenn Replikate eines Objekts in verschiedenen Teilnetzen sind und Änderungen durchgeführt werden sollen, können nicht alle Replikate erreicht werden und dadurch inkonsistente Zustände entstehen.

Zur Lösung dieses Problems gibt es u.a. die sog. *Quorum-Protokolle*, bei denen jedem Replikat vorher ein bestimmtes Gewicht (eine Anzahl von Stimmen) zugeordnet wird. Um eine Operation ausführen zu können, muss eine bestimmte Summe von Gewichten gesammelt werden, d.h. zum Lesen müssen r Stimmen, zum Schreiben w Stimmen vorhanden sein (Read-Quorum bzw. Write-Quorum).

In einem System seien N Replikate vorhanden. Jedem Replikat sei ein Gewicht von 1 zugeordnet. Begründen Sie für die vier folgenden Quorum-Protokolle, ob sie ein sinvolles Lese- bzw. Schreib-Quorum enthalten.

a) $N = 10, r = 6, w = 5$

b) $N = 10, r = 3, w = 7$

c) $N = 10, r = 1, w =$ N

d) $N = 10, r = 5, w = 6$

Aufgabe 4: Quorum-basierte Replikation

Eine Datei wird auf zehn Servern repliziert. Nennen Sie alle Kombinationen von Lese-Quorum und Schreib-Quorum, die durch den Abstimm-Algorithmus erlaubt sind.

Aufgabe 5: Programmierung: Konsistenzprotokolle

In der Vorlesung (Kapitel 8.5) haben Sie verschiedene Konsistenzprotokolle kennen gelernt, dazu gehören auch die *primary-based*-Protokolle. In *primary-based*-Protokollen ist jedem Datenelement x im Datenspeicher ein primärer Server zugeordnet, der dafür verantwortlich ist, Schreiboperationen für x zu koordinieren. Ausserdem kann unterschieden werden, ob die primäre Kopie auf einem fest vorgegebenen entfernten Server liegt, oder ob Schreiboperationen lokal ausgeführt werden können, indem die primäre Kopie jeweils auf den Server verschoben wird, der die Schreiboperation initiiert hat.

a) Implementieren Sie auf Basis des gegebenen Codes im Archiv `u11Files.zip`[1] auf der Vorlesungswebseite ein *remote-write*-Protokoll mit Java RMI. Erweitern Sie dazu die Methode `write()` und führen Sie ggf. neue Remote-Methoden für das Protokoll ein. Denken Sie dabei daran, daß bei RMI der Aufruf einer Remote-Methode immer eine Anfrage an den entfernten Server sowie eine zugehörige Antwort (Bestätigung) des Servers zur Folge hat.

Der vorgegebene Client verbindet sich mit einem Server (ausgewählt über das Kommandozeilenargument) und führt eine Schreib- und eine Leseoperation durch. Starten Sie zum Test möglichst gleichzeitig mehrere Instanzen des Clients, die sich jeweils mit anderen Servern verbinden. Testen Sie Ihre Lösung mit einer unterschidlichen Anzahl von Servern.

b) Überlegen Sie, wie sie ein *local-write*-Protokoll implementieren würden. Was ist die Schwierigkeit dabei?

Aufgabe 6: Noch Fragen?

Nutzen Sie die Gelegenheit und bereiten Sie Fragen zu den Übungsblättern vor, die Sie in der Übungsstunde diskutieren möchten. Hier können jetzt Schwierigkeiten beim Lösen der Aufgaben besprochen und Lösungswege erneut erläutert werden.

[1] `http://www.bs.informatik.uni-siegen.de/web/wismueller/vl/vs/u11Files.zip`