

Institut für Betriebssysteme und Rechnerverbund
Übungen zur Vorlesung “Verteilte Systeme”, WS 02/03

<http://www.ibr.cs.tu-bs.de/lehre/ws0203/vs/>

Dozent: Prof. Dr. Stefan Fischer <fischer@ibr.cs.tu-bs.de> · Übungsleiter: Frank Strauß <strauss@ibr.cs.tu-bs.de>

11 Replikation

Übung am 29.01.2003

11.1 Majority Consensus

Quorum-basierte Protokolle (Votierungsverfahren) (Folie 11-39) stellen einen Kompromiss zwischen Ausfallsicherheit und Performanz bei replizierenden Systemen dar. Dabei dienen Mehrheitsentscheidungen (*Majority Consensus*) dazu, darüber zu entscheiden, ob gewünschte Lese- oder Schreiboperationen ausgeführt werden können, ohne die Konsistenz zu verletzen. Dies sind die zwei Regeln, die bei der Parametrisierung der Quoren — also der Anzahlen der benötigten Stimmen, um eine Read- bzw. Write-Operation durchzuführen — zu beachten sind:

$$LQ + SQ > S \quad (1)$$

$$SQ > \frac{S}{2} \quad (2)$$

Wenn einer der S Server einen Request für eine Lese- oder Schreiboperation erhält, so schickt er eine Votierungs-Nachricht an alle Server (inkl. sich selbst). Jeder Server, der seine Stimme nicht bereits vergeben hat, gibt dem anfragenden Server seine Stimme (zusammen mit der Versionsnummer). Der initiiierende Server ermittelt daraus die Anzahl der erhaltenen Stimmen und das Maximum der Versionsnummern¹.

Hat der Server im Falle einer Lese-Operation mindestens LQ (Lese-Quorum) Stimmen erhalten, so besorgt er sich von dem zustimmenden Server, der die größte Versionsnummer hat, das Datum und führt damit die Lese-Operation aus.

Im Falle einer Schreib-Operation benötigt der initiiierende Server mindestens SQ (Schreib-Quorum) Stimmen. Sofern er diese erhalten hat, kann er die Schreib-Operation erfolgreich durchführen, indem er das neue Datum an alle Server sendet, deren Stimme er bekommen hat. Dabei wird die Versionsnummer auf all diesen Servern auf $v_{max} + 1$ erhöht.

Nach Abschluss der Operation werden allen beteiligten Servern ihre Stimmen zurückgegeben.

- (a) Welche zwei Ereignisse, die die Konsistenz verletzen könnten, schließt Regel (1) aus?
- (b) Welches weitere Ereignis, das die Konsistenz verletzen könnte, schließt Regel (2) aus?
- (c) Warum sind die hier angegeben zwei Regeln flexibler als die Regel aus der Vorlesung

$$LQ = SQ = \frac{N}{2} + 1$$

- (d) Offensichtlich gilt es, einen Kompromiss zwischen kleinem LQ und kleinem SQ zu finden. Diskutieren Sie Anwendungsfälle, in denen kleine/große LQ/SQ sinnvoll sind.

¹Das Verfahren kann variieren: Man kann per Multicast um Stimmen bitten und zuviel erhaltene Stimmen sofort wieder zurückgeben. Es ist aber auch möglich, einzelne Stimmen einzusammeln und bei Erreichen der erforderlichen Anzahl abzubrechen.

- (e) Warum sollte man sich über die *Granularität* der Daten Gedanken machen? Beispiel: Warum ist es besser, Abstimmungen über die Zugriffe auf einzelne Dateien auszuführen, anstatt über Zugriffe auf das gesamte Dateisystem zu entscheiden?
- (f) Exemplarisch betrachten wir ein Netz mit $S = 7$ Servern und $LQ = 3$ und $SQ = 5$. Die Replike des einzigen im folgenden benutzten Datenobjektes haben auf den Servern anfangs die angegebenen Versionsnummern. Skizzieren Sie in dem Diagramm auf der folgenden Seite den (bzw. einen möglichen) Verlauf der Nachrichten, die durch die angegebenen Operationen ausgelöst werden, indem Sie für Votierungs-, Stimmen-, Freigabe- und Datenübertragungs-Nachrichten entsprechende Pfeile einzeichnen (ggf. mit Versionsnummer).
- (g) Welche Operationen wären im weiteren Verlauf im Falle einer 3-zu-4 Partionierung in den beiden Netzpartitionen noch möglich? Welche im Falle einer 2-zu-5 Partionierung?

