TU München, Fakultät für Informatik
Lehrstuhl III: Datenbanksysteme
Prof. Alfons Kemper, Ph.D.

Übung zur Vorlesung *Einsatz und Realisierung von Datenbanken* im SoSe23

Alice Rey, Maximilian Bandle, Michael Jungmair (i3erdb@in.tum.de)
http://db.in.tum.de/teaching/ss23/impldb/

Blatt Nr. 06

Hinweise Machen sie sich mit Raft vertraut: `http://thesecretlivesofdata.com/raft/`.

Hausaufgabe 1

Zeigen Sie, dass die *write-all / read-any* Methode zur Synchronisation replizierter Daten einen Spezialfall der *Quorum-Consensus*-Methode darstellt.

- Für welche Art von Workloads eignet sich dieses Verfahren besonders gut?
- Wie werden Stimmen zugeordnet um *write-all / read-any* zu simulieren?
- Wie müssen die Quoren Q_w und Q_r vergeben werden?

Dieses Verfahren fordert einen sehr großen Aufwand beim Schreiben, aber nur minimalen Aufwand beim Lesen. Daher eignet es sich besonders gut für Workloads in denen wesentlich mehr Daten gelesen als geschrieben werden. Dies entspricht z.B. analytischen Anfragen.
Siehe Übungsbuch.

Hausaufgabe 2

Um Ausfallsicherheit zu garantieren, ist ein Datenwert 'A' auf vier Rechnern verteilt. Jeder Rechner hält dabei eine vollständige Kopie von 'A'. Um Konsistenz zu garantieren, wird das Quorum-Consensus-Verfahren eingesetzt. Dabei ist jedem Rechner ein Gewicht $w_i(A)$ wie folgt zugewiesen:

Rechner	Kopie	Gewicht
R_1	A_1	3
R_2	A_2	1
R_3	A_3	2
R_4	A_4	2

Das Lesequorum ist $Q_r(A) = 4$ und das Schreibquorum is $Q_w(A) = 5$.

a) Geben Sie **alle** Lesemöglichkeiten für eine Transaktion auf dem Datum 'A' nach dem Quorum-Consensus-Protokoll an.
A_1,A_2
A_1,A_2,A_3
A_1,A_2,A_3,A_4
A_1,A_2,A_4
A_1,A_3
A_1,A_3,A_4
A_1,A_4
A_2,A_3,A_4
A_3,A_4

b) Geben Sie **alle** Schreibmöglichkeiten für eine Transaktion auf dem Datum 'A' nach dem Quorum-Consensus-Protokoll an.

A_1,A_2,A_3
A_1,A_2,A_3,A_4
A_1,A_2,A_4
A_1,A_3
A_1,A_3,A_4
A_1,A_4
A_2,A_3,A_4

c) Zeigen Sie für dieses Beispiel, dass, während eine Transaktion T_1 ein Schreibquorum auf A hält, es für andere Transaktionen T_x nicht möglich ist, ein Lesequorum für A zu bekommen.

Zum Schreiben muss die Transaktion T_1 mindestens Kopien mit einem Gesamtgewicht von 5 gesperrt haben. Insgesamt haben alle Kopien zusammen das Gewicht 8. Somit können maximal Kopien mit einem Gewicht von zusammen 3 übrig bleiben, womit das Lesequorum von 4 nicht mehr erfüllt werden kann.

Hausaufgabe 3

Gegeben seien die Tabellen `Studenten` und `Punkte` mit Schlüssel `MatrNr`, wobei `Punkte` auf einem separaten Rechner gespeichert ist. Es soll folgende Anfrage ausgeführt werden:

```
SELECT Name, Bonus FROM Student s, Punkte p WHERE s.MatrNr = p.MatrNr;
```

Der Datenbankadministrator entscheidet sich für einen Bloom-Filter zur Vorauswahl der Tupel. Auf `MatrNr` wird die Hash-Funktion $h(x) = x \mod 5$ angewendet.

`Studenten`

MatrNr	**Name**	**Hashwert**
27	Magda	2
4	Josef	4
19	Erik	4
95	Philipp	0

`Punkte`

MatrNr	**Bonus**	**Hashwert**
27	ja	2
16	nein	1
25	nein	0
95	ja	0

a) Berechnen Sie die Hash-Werte und tragen Sie diese in die obige Tabelle ein.

b) Geben Sie den von `Studenten` zu übertragenden Bitvektor an.

Bitvektor: 10101
$h(x)$ hat fünf verschiedene Ausgaben. D.h. Vektor ist min. 5 Bits lang. Ein Bit wird dann gesetzt, wenn die Hashfunktion es einmal ausgegeben hat.

c) Geben Sie basierend auf dem Bitvektor an, welche Tupel aus `Punkte` übertragen werden.

27, 25, 95

d) Geben Sie die Falsch-Positiv-Rate (false positive rate) an.

33%

e) Nehmen Sie an, dass jedes Tupel 8 Byte und der Bloomfilter selbst 1 Byte groß ist. Berechnen Sie zunächst die übertragenen Bytes ohne und mit Einsatz des Bloom-Filters.
Ohne Filter: $4 \cdot 8 = 32$
Mit Filter: $3 \cdot 8 + 1 = 25$

Hausaufgabe 4

Um ein Gefühl für das *Raft Consensus Protokoll* zu bekommen, führen Sie folgende Aktionen mit *RaftScope* unter `https://raft.github.io/` aus. Die Simulation kann mit einem Klick auf das Uhr Symbol gestoppt und gestartet werden. Ein Rechtsklick auf einen der Server öffnet ein Menü um Aktionen auszulösen.

Führen Sie folgende Aktion aus und beschreiben Sie in Stichpunkten was passiert.

1. Warten sie bis die erste Leader Election abgeschlossen ist.
2. Senden Sie einen Request an den Leader. (Rechtsklick auf Leader)
3. Stoppen Sie den Leader (Server ausschalten), warten Sie bis ein neuer Leader gewählt wurde, und senden Sie einen Request an den neuen Leader.
4. Wiederholen Sie den vorigen Schritt 2x bis nur noch 2 Knoten übrig sind. Wird ein neuer Leader gewählt?
5. Starten Sie (resume) wieder einen weiteren Server. Wird ein neuer Leader gewählt?

Lösung:

1. Am Ende der ersten Leader Election (z.B. von S2) sind alle Knoten blau markiert und in der zweiten Phase.

2. Der Leader hängt den Request in seinem Log an und leitet ihn an alle Follower weiter. Alle erreichbaren Follower merken sich den Request im Log vor (gestrichelte Linie) und bestätigen diesen. Wenn die Mehrheit bestätigt, updated der Leader seinen Status. In der nächsten Hearbeat-Nachricht teilt der Leader das Update seines Status allen Followern mit. Diese setzen den vorgemerkten Log auf gültig (durchgezogene Linie im Log).

3. Der gestoppte Leader sendet keine Hearbeat-Nachrichten mehr. Dadurch läuft der Timeout der Follower weiter und die ersten Follower-Knoten, bei denen der Timer abläuft werden zu Candidates. Dies löst eine neue Leader Election durch die Mehrheit aller Knoten aus. Der neu gewählte Leader verarbeitet den Request wie oben beschrieben.

4. Beim zweiten gestoppten Knoten verläuft das Stoppen des Leaders wie in der vorigen Aufgabe. Beim dritten gestoppten Knoten ist nicht mehr die Mehrheit der Knoten online und es kann kein neuer Leader gewählt werden. Die verbleibenden Knoten versuchen weiter als Candidate genug Stimmen zu erhalten. Es kann kein neuer Leader gewählt werden.

5. Der erste Candidate der einen Timeout hat, fragt alle Knoten nach einer Stimme und bekommt die nötige Mehrheit. Mit dieser Nachricht wird die Phase des neu gestarteten Knotens aktualisiert. Der Candidate wird als neuer Leader gewählt.

Hausaufgabe 5

Beantworten Sie folgende Fragen zum RAFT Protokoll. Verwenden Sie *RaftScope* unter `https://raft.github.io/` um Ihre Vermutungen zu bestätigen.

1. Wie viele Server müssen ausfallen, dass in einem Cluster mit n Servern kein neuer Leader bestimmt werden kann?
2. Wie können Sie mit `restart` und `time-out` in RaftScope einen bestimmten Knoten als neuen Leader erzwingen? Geben Sie die Schritte an.
3. Wie verläuft das Beispiel in Abbildung 1 weiter? Wie kann sicher gestellt werden, dass der neu gewählte Leader den neuesten Logeintrag hält? Beschreiben Sie in Stichpunkten.

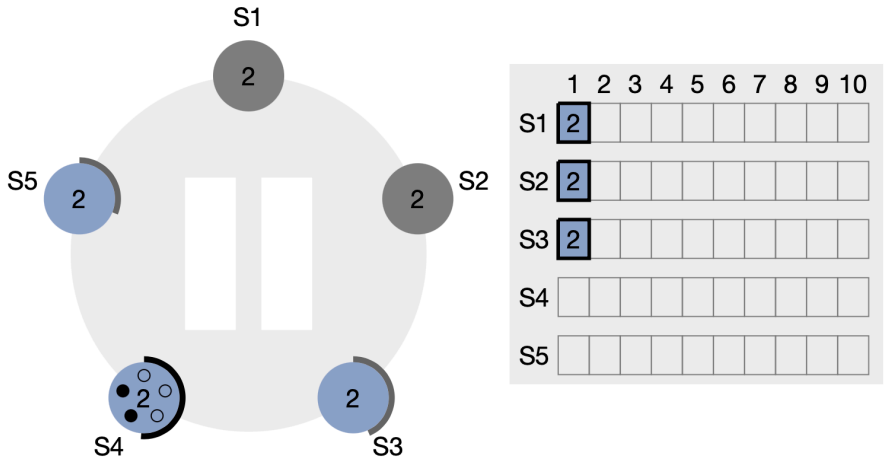

Abbildung 1: Beispiel eines Status in RaftScope

Lösung:

1. Es müssen mindestens $\lceil n/2 \rceil$ Server ausfallen.
2. Folgende Schritte sind nötig um einen Leader zu erzwingen:
   a) Starten sie den aktuellen Leader mit `restart` neu. Er started als Follower.
   b) Nutzen Sie `time-out` um beim gewünschten Knoten den Timout abzubrechen.
   c) Der Knoten wird Candidate und holt sich Stimmen ein und wird neuer Leader.

   Dies zeigt, dass der Knoten der zufällig den geringsten Time-Out hat der neue Leader wird.
3. Folgende Schritte verlaufen als nächstes in Abbildung 1.
   a) S4 ist ein Candidate, aber der Log von S4 enthält nicht den letzten comitteden Log-Eintrag.
   b) S3 stimmt nicht für S4, weil dessen eigener Log *aktueller* ist.
   c) S4 wird nicht Leader, weil die Mehrheit der Knoten einen *akutelleren* Logeintrag als S4 haben.
   d) Irgendwann hat ein anderer Knoten, zum Beispiel S3 (mit dem neusten committeden Log-Eintrag), zuerst einen Timeout haben und wird neuer Candidate.
   e) Die anderen Knoten wählen S3, weil S3 den *aktuellsten* Log-Zustand hat und somit wird S3 zum neuen Leader.
   f) S3 stellt als neuer Leader sicher, dass alle Knoten den *aktuellsten* Logeintrag haben.
   g) Dieses Szenario funktioniert immer, denn:
      i. Die Mehrheit der Server enthalten den zuletzt committeden Log-Eintrag.
      ii. Die Mehrheit der Server werden benötigt um einen neuen Leader zu wählen.

      Beide Mehrheiten des selben Clusters müssen sich zur erfolgreichen Wahl des Leaders überschneiden. Irgendwann wird ein neuer Leader mit dem neuesten committeden Log-Eintrag gewählt.

Schritte um das Beispiel aus der 3. Teilaufgabe in *RaftScope* zu simulieren:

1. Starten Sie die Simulation und stoppen Sie direkt S4.
2. Warten Sie die Wahl des Leaders ab. Forcieren sie einen Leader, der nicht S5 ist, mittels Timeout, falls notig.
3. Stoppen Sie S5.
4. Stellen Sie einen request an den Leader und warten Sie ab.
5. Stoppen Sie S1 und S2 und starten Sie S4 und S5.
6. Forcieren Sie mittels Timout, dass S5 Candidate wird.