Computergestützte Gruppenarbeit

Übungsblatt 2

21.06.2005

Dr. Jürgen Vogel

European Media Laboratory (EML)
Heidelberg

SS 2005

Aufgabe 1: Soft State-Verfahren (1)

Implementieren Sie das in der Vorlesung besprochene Soft State-Verfahren anwendungsunabhängig in Pseudo-Code:

- verwenden Sie für die Verwaltung des Anwendungszustands die folgende Datenstruktur:

```
Type Object {
    Integer id              // eindeutiger Identifizierer
    Binary data             // Anwendungsdaten
    Boolean active          // lokal aktiv?
    Time announced          // letzte Ankündigung
}
Hashtable Object objects    // Hashtabelle mit allen Objekten
                            // Zugriff auf Objekte per Iterator
                            // objects[0],…,objects[n] oder auf
                            // einzelne Objekte per objects(id)
```

- schreiben Sie die folgenden Funktionen
  - `setInterest(Integer id, Boolean active)`
    Wird von der Anwendung aufgerufen, um das Objekt mit dem Identifizierer id aktiv/passiv zu setzen. Jede Instanz kündigt nur die Objekte an, die lokal aktiv sind.

Aufgabe 1: Soft State-Verfahren (2)

- `changeData(Integer id, Binary data)`
  Wird von der Anwendung zur Veränderung des Objekts id aufgerufen.
- `update()`
  Wird periodisch alle T Zeiteinheiten aufgerufen und versendet die lokalen Ankündigungen. Verwenden Sie zu diesem Zweck die Funktion `send(Integer id)`.
- `receive(Integer id, Binary data)`
  Wird von der Netzwerkschnittstelle aufgerufen, wenn die Ankündigung einer entfernten Instanz zum Objekt id empfangen wurde.

- Objekte, die seit 5T nicht angekündigt wurden, sollen mit der Funktion `delete(Integer id)` gelöscht werden.
- Zum Abfragen der aktuellen Zeit können sie die Funktion `getTime()` verwenden.

Aufgabe 2: Hard State vs. Soft State (1)

Beantworten Sie anhand des angegebenen Artikels die folgenden Fragen.

P. Ji, Z. Ge, J. Kurose and D. Towsley, A Comparison of Hard-state and Soft-state Signaling Protocols, In: Proc. ACM SIGCOMM, Karlsruhe, Germany, August 2003

1) Wieso eignet sich das Soft State-Verfahren insbesondere bei verlustbehafteter Datenübertragung gut?

2) Benötigt man beim Soft State- bzw. Hard State-Verfahren eine Absicherung gegen den unkontrollierten Ausfall von Instanzen, wenn jede Instanz für bestimmte Zustandsteile zuständig ist?

3) Was versteht man beim Soft State-Verfahren unter *fälschlichem Löschen* ("false removal")?

4) Was verstehen die Autoren unter einer "Trigger"-Nachricht?

Aufgabe 2: Hard State vs. Soft State (2)

5) Wie messen die Autoren eine Inkonsistenz und deren Dauer? Welche Inkonsistenzen können beim reinen Soft State-Verfahren auftreten und wodurch werden diese repariert?

6) Welche Erweiterungen des reinen Soft State-Verfahrens schlagen die Autoren vor und warum?

7) Diskutieren Sie den Trade-off bei der Wahl der Ankündigungs-periode T.

8) Von welchen Faktoren wird die Performance des Signalisierungs-Algorithmus beeinflusst?

9) Vergleichen Sie die Verfahren HS und SS+RTR.

10) Leiten Sie anhand von Abbildung 4 Empfehlungen für die Verwendung der verschiedenen Signalisierungsverfahren ab.

Aufgabe 3: Operations-Transformation (1)

In einer synchronen Sitzung bearbeiten drei Benutzer einen Text mit dem Anfangszustand S_0 = "ABCDEFGH". Die Kausalitäts-überprüfung soll per Zustandsvektor durchgeführt werden (SV_{S_0} = $\langle(i,0),(j,0),(k,0)\rangle$) und die Intentionserhaltung mit Operations-Tranformation. Die Instanzen-ID's haben folgenden Prioritäten: $i < j < k$.

Gegeben sei der folgende Zeitablauf mit den Operationen

- O_1 = "lösche von Index 2 bis Index 4"
- O_2 = "füge 'abcd' bei Index 4 ein"
- O_3 = "lösche von Index 5 bis Index 8"
- O_4 = "lösche von Index 6 bis Index 7"

Bestimmen Sie

- die Zustandsvektoren aller Operationen und Zwischenzustände
- die Intention aller Operationen
- alle benötigten Transformationsschritte
- den Endzustand bei allen Instanzen

Aufgabe 3: Operations-Transformation (2)