

Übungen zur Vorlesung
Mobile und Verteilte Datenbanken
 WS 2008/2009

Übung 13 – Replikation in Sensornetzwerken

1. Aufgabe: Datenspeicherung in Netzwerken

Gegeben ist ein Netzwerk bestehend aus N Knoten, 1 Datensenke und einem Durchmesser von \sqrt{N} . Die Koordinaten der Datensenke sind bekannt. Das Senden eines Events von einem Knoten zur Senke unter Verwendung von geographischem Routing verbraucht $O(\sqrt{N})$ Nachrichten. Das Fluten des Netzwerkes kostet hingegen $O(N)$ Nachrichten.

Basierend auf diesen Informationen, füllen Sie die folgende Tabelle für die Speicherung und Abfrage eines Events aus:

	Externe Speicherung	Lokale Speicherung	Datenzentrische Speicherung
Speicherung	$O(\sqrt{N})$	--	$O(\sqrt{N})$
Anfrage	--	$O(N)$	$O(\sqrt{N})$
Antwort	--	$O(\sqrt{N})$	$O(\sqrt{N})$

2. Aufgabe: REALM

Gehen Sie das Beispiel aus der Vorlesung zum Replica Allocation Algorithm für den Server S1 mit folgender Access Log Tabelle durch.

Access Log für Server 1

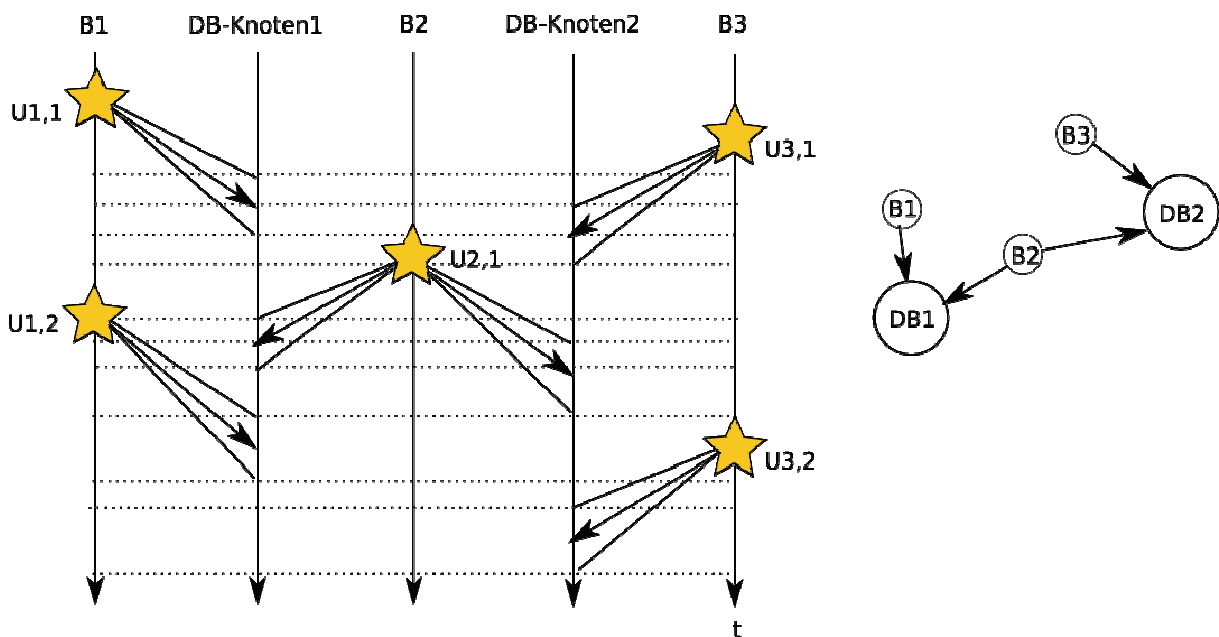
Daten	Mobile Hosts	Anzahl Zugriffe
D1	C4	23
D1	C3	17
D2	C5	5
D6	S2	15
D3	C5	9

- 1) Zugriffshäufigkeiten AF
 - a) AF $S1 \rightarrow D1$: 40 (C4,C3)
 - b) AF $S1 \rightarrow D2$: 5 (C5)
 - c) AF $S1 \rightarrow D6$: 15 (S2)
 - d) AF $S1 \rightarrow D3$: 9 (C5)
- 2) geordnete Liste: $L = \{D1, D6, D3, D2\}$
- 3) Füge interessante Daten hinzu:
 - a) D5 (interessant für C5 und C3)
 - b) D4 (interessant für S2)
 - c) $\rightarrow L = \{D1, D6, D3, D2, D5, D4\}$
- 4) Lösche Daten, die durch andere Server bereits gespeichert werden
 - a) S2 speichert D2, S3 speichert D3, S1 speichert D1

- b) $\rightarrow L=\{D6,D5,D4\}$
 5) Speichere Daten auf anderen Servern
 a) S2 nutzt D6 \rightarrow S2 sollte D6 speichern
 b) $L = \{D5,D4\}$
 6) Speichere D5 und D4

3. Aufgabe: Update-Linearisierung

Gegeben sind 3 Beobachter und 2 DB-Knoten in einem Netzwerk. Die Abbildung zeigt eintreffende Aktualisierungsanforderungen für ein Events bei den DB-Knoten. Wenden Sie den in der Vorlesung vorgestellten Algorithmus an, um die Events geordnet zu replizieren. Geben Sie alle wichtigen Zwischenschritte (Add und Join) und die Ordnungsgraphen der beiden DB-Knoten an. Die Join-Nachricht nach Update U1,1 kommt erst nach dem Update U3,1 beim DB-Knoten 2 an. Gleiches gilt für Update U2,1.



- 1) DB-Knoten 1 Add $\{U1,1\}$
- 2) DB-Knoten 2 Add $\{U3,1\}$
 - a. DB-Knoten 2 Join $\{U3,1\}$ U1,1 nicht akzeptiert
 - b. DB-Knoten 1 Join $\{U1,1\}$ U3,1 nicht akzeptiert
- 3) DB-Knoten 1 Add $\{U1,1 \rightarrow U2,1\}$
- 4) DB-Knoten 2 Add $\{U3,1 \rightarrow U2,1\}$
 - a. DB-Knoten 2 Join $\{U1,1 \rightarrow U2,1\}; \{U3,1 \rightarrow U2,1\}$
 - b. DB-Knoten 1 Join $\{U3,1 \rightarrow U2,1\}; \{U1,1 \rightarrow U2,1\}$
- 5) DB-Knoten 1 Add U1,2 $\{U3,1 \rightarrow U2,1 \rightarrow U1,2\}$
 - a. DB-Knoten 2 Join $\{U3,1 \rightarrow U2,1 \rightarrow U1,2\}$
- 6) DB-Knoten 2 Add U3,2 $\{U2,1 \rightarrow U1,2 \rightarrow U3,2\}$
 - a. DB-Knoten 1 Join $\{U2,1 \rightarrow U1,2 \rightarrow U3,2\}$