

Übungen zur Vorlesung  
**Mobile und Verteilte Datenbanken**

WS 2008/2009

**Lösung Blatt 14**

**Aufgabe 1: Beispiel-Ablauf 2PC**

- a) Skizzieren Sie Abarbeitung einer Transaktion im 2PC Protokoll mit 3 Teilnehmern für den Commit-Fall. Geben Sie die Anzahl der übertragenen Nachrichten an und gehen Sie dabei von verlässlichem Broadcast in einer Single-Hop Umgebung aus.

**Lösung:**

Im Commit-Fall wird gesendet:

- 1 prepared (vom „Startknoten“)
- 2 prepare (vom Koordinator)
- 3 prepared (von den übrigen Teilnehmern, also hier 2x)
- 4 commit (vom Koordinator)

Es werden also 5 Nachrichten gesendet (4 falls sich der Koordinator-Prozess auf dem Startknoten befindet)

- b) Wann kann es zu einer Blockierung kommen?

**Lösung:**

Es kommt zu einer Blockierung, wenn ein Knoten für Commit gestimmt hat, aber keine Nachricht mehr vom Koordinator erhält.

**Aufgabe 2: Beispiel-Ablauf Paxos Commit**

- a) Skizzieren Sie analog zu Aufgabe 1 die Abarbeitung einer Transaktion im Paxos Commit Protokoll mit 3 Teilnehmern für den Commit-Fall und geben Sie wieder die Anzahl der übertragenen Nachrichten an.

**Lösung:**

Im Commit-Fall wird gesendet:

- 1 beginCommit (vom Startknoten)
- 2 prepare (vom Initial Leader)
- 3 acceptPrepare (von allen Teilnehmern)
- 4 acceptedPrepare (von allen Akzeptoren)
- 5 commit (vom Initial Leader)

Es werden also  $1+1+3+3+1 = 9$  Nachrichten gesendet, wenn man berücksichtigt, dass mehrere Prozesse auf einem Knoten laufen noch  $1+1+2+2+1 = 7$  Nachrichten.

### Aufgabe 3: Beispiel-Ablauf CLCP

- a) Wiederholen Sie die Skizzierung der Abarbeitung einer Transaktion im CLCP mit 3 Teilnehmern für den Commit-Fall. Geben Sie dabei die Anzahl der übertragenen Nachrichten und den Inhalt der jeweiligen Commit-Matrizen an.

#### Lösung:

1. Schritt: P1 sendet

CM1		1		2		3
	1	voteCommit				
	2					
	3					

2. Schritt: P2 und P3 senden

CM2		1		2		3	CM3		1		2		3
	1	voteCommit		voteCommit				1	voteCommit				voteCommit
	2			voteCommit				2					
	3							3					voteCommit

3. Schritt: nach Empfang von CM2 und CM3 kann P1 bereits committen:

CM1		1		2		3
	1	voteCommit		voteCommit		voteCommit
	2	voteCommit		voteCommit		
	3	voteCommit				voteCommit

Bei P2 und P3 ist noch nicht in allen Zeilen eine Mehrheit für voteCommit vorhanden:

CM2		1		2		3	CM3		1		2		3
	1	voteCommit		voteCommit				1	voteCommit				voteCommit
	2			voteCommit				2			voteCommit		voteCommit
	3			voteCommit		voteCommit		3					voteCommit

4. Schritt: nun sind alle Felder der drei Matrizen voteCommit

- b) Wie sieht der Ablauf aus, wenn Teilnehmer 2 und 3 keine Nachrichten von Teilnehmer 1 empfangen können?

**Lösung:** Es kommt erst zu timeOut, dann zu timeOutAck und dann zu Abort, also:

Schritt 1: P1 würde senden

CM1		1		2		3
	1	voteCommit				
	2					
	3					

Und ebenso P2 und P3:

CM2		1		2		3	CM3		1		2		3
	1							1					
	2			voteCommit				2					
	3							3					voteCommit

Schritt 2: P2 und P3 erhalten keine Nachricht von P1

CM2		1		2		3	CM3		1		2		3
	1							1					
	2			voteCommit				2			voteCommit		voteCommit
	3			voteCommit		voteCommit		3					voteCommit

Schritt 3:

CM2	1	2	3	CM3	1	2	3
1					1		
2		voteCommit	voteCommit		2	voteCommit	voteCommit
3		voteCommit	voteCommit		3	voteCommit	voteCommit

Schritt 4: Irgendwann werden jeweils Timeouts geworfen..

CM2	1	2	3	CM3	1	2	3
1		voteTimeOut	voteTimeOut		1	voteTimeOut	voteTimeOut
2		voteCommit	voteCommit		2	voteCommit	voteCommit
3		voteCommit	voteCommit		3	voteCommit	voteCommit

Schritt 5: .. und von dem jeweils anderen Knoten bestätigt,

CM2	1	2	3	CM3	1	2	3
1		timeOutAck	timeOutAck		1	timeOutAck	timeOutAck
2		voteCommit	voteCommit		2	voteCommit	voteCommit
3		voteCommit	voteCommit		3	voteCommit	voteCommit

Was dann zum Abort der Transaktion führt.

#### Aufgabe 4: Overhead der Commit-Matrizen

Mit steigender Nachrichtengröße steigt in drahtlosen Netzen auch die Wahrscheinlichkeit von Kollisionen auf dem MAC-Layer, die eine erneute Übertragung einer Nachricht erforderlich machen. Die Nachrichtengröße wird daher häufig auf etwa 50 Byte begrenzt.

Wieviele Teilnehmer können bei einer geeigneten Kodierung der Commit-Matrizen maximal an einer Transaktion teilnehmen, wenn eine Matrix in einer einzigen, maximal 50 Byte großen Nachricht übertragen werden soll?

#### Lösung:

Die 5 möglichen Zustände in einer Commit-Matrix lassen sich mit jeweils 3 Bit kodieren, mehr Informationen sind nicht nötig, da die Anzahl der Teilnehmer vorher bekannt sind. Aus  $50 \text{ Byte} > (3 \cdot p^2) / 8$  ergibt sich  $11,54 > p$ , ohne zusätzliche Komprimierung können also höchstens 11 Knoten an einer Transaktion teilnehmen.