VerteilteWebInf Hausaufgabe 9

Gruppe 6

December 17, 2014

Aufgabe 1

In der folgenden Tabelle werden pro Zeitschritt jeweils die Knoten aufgeführt, bei denen sich der Status im Gegensatz zum vorherigen Zeitschritt geändert hat. Knoten, die in einem Wartezustand sind (WARTE in einem Status) sind in jedem Zeitschritt aufgeführt.
Der angegebene Status beschreibt den Status nach Verarbeitung des Inputs.

t	K	Status	Input	Entscheidung	Output	Kommentar
1	C	$UPDATE_{T1}(x)$ OK	-	-	$C : UPDATE_{T1}(x)$ OK	
2	D	$UPDATE_{T1}(x)$ OK	$C : UPDATE_{T1}(x)$ OK	-	$C : UPDATE_{T1}(x)$ OK $D : UPDATE_{T1}(x)$ OK	
	A	$UPDATE_{T2}(x)$ OK	-	-	$A : UPDATE_{T2}(x)$ OK	
3	E	$UPDATE_{T1}(x)$ OK	$C : UPDATE_{T1}(x)$ OK $D : UPDATE_{T1}(x)$ OK	-	$C : UPDATE_{T1}(x)$ OK $D : UPDATE_{T1}(x)$ OK $E : UPDATE_{T1}(x)$ OK	
	B	$UPDATE_{T2}(x)$ OK	$A : UPDATE_{T2}(x)$ OK	-	$A : UPDATE_{T2}(x)$ OK $B : UPDATE_{T2}(x)$ OK	
	F	$UPDATE_{T3}(x)$ OK	-	-	$F : UPDATE_{T3}(x)$ OK	
4	F	$UPDATE_{T3}(x)$ OK $UPDATE_{T1}(x)$ PASSIERE	$C : UPDATE_{T1}(x)$ OK $D : UPDATE_{T1}(x)$ OK $E : UPDATE_{T1}(x)$ OK	-	$C : UPDATE_{T1}(x)$ OK $D : UPDATE_{T1}(x)$ OK $E : UPDATE_{T1}(x)$ OK $F : UPDATE_{T1}(x)$ PASSIERE	
	C	$UPDATE_{T1}(x)$ OK $UPDATE_{T2}(x)$ WARTE	$A : UPDATE_{T2}(x)$ OK $B : UPDATE_{T2}(x)$ OK	-	-	Warte, weil Antrag (T_2) mit niedriger priorisiertem Antrag (T_1) in Konflikt steht
	G	$UPDATE_{T3}(x)$ OK	$F : UPDATE_{T3}(x)$ OK	-	$F : UPDATE_{T3}(x)$ OK $G : UPDATE_{T3}(x)$ OK	

t	K	Status	Input	Entscheidung	Output	Kommentar
5	G	$UPDATE_{T3}(x)$ OK $UPDATE_{T1}(x)$ PASSIERE	$C : UPDATE_{T1}(x)$ OK $D : UPDATE_{T1}(x)$ OK $E : UPDATE_{T1}(x)$ OK $F : UPDATE_{T1}(x)$ PASSIERE	-	$C : UPDATE_{T1}(x)$ OK $D : UPDATE_{T1}(x)$ OK $E : UPDATE_{T1}(x)$ OK $F : UPDATE_{T1}(x)$ PASSIERE $G : UPDATE_{T1}(x)$ PASSIERE	
	C	$UPDATE_{T1}(x)$ OK $UPDATE_{T2}(x)$ WARTE	-	-	-	
	A	$UPDATE_{T2}(x)$ OK $UPDATE_{T3}(x)$ WARTE	$F : UPDATE_{T3}(x)$ OK $G : UPDATE_{T3}(x)$ OK	-	-	
6	A	$UPDATE_{T2}(x)$ OK $UPDATE_{T3}(x)$ WARTE $UPDATE_{T1}(x)$ PASSIERE	$C : UPDATE_{T1}(x)$ OK $D : UPDATE_{T1}(x)$ OK $E : UPDATE_{T1}(x)$ OK $F : UPDATE_{T1}(x)$ PASSIERE $G : UPDATE_{T1}(x)$ PASSIERE	-	$C : UPDATE_{T1}(x)$ OK $D : UPDATE_{T1}(x)$ OK $E : UPDATE_{T1}(x)$ OK $F : UPDATE_{T1}(x)$ PASSIERE $G : UPDATE_{T1}(x)$ PASSIERE $A : UPDATE_{T1}(x)$ PASSIERE	
	C	$UPDATE_{T1}(x)$ OK $UPDATE_{T2}(x)$ WARTE	-	-	-	
7	B	$UPDATE_{T2}(x)$ OK $UPDATE_{T3}(x)$ WARTE $UPDATE_{T1}(x)$ PASSIERE	$C : UPDATE_{T1}(x)$ OK $D : UPDATE_{T1}(x)$ OK $E : UPDATE_{T1}(x)$ OK $F : UPDATE_{T1}(x)$ PASSIERE $G : UPDATE_{T1}(x)$ PASSIERE $A : UPDATE_{T1}(x)$ PASSIERE	ABORT T_1	@ALL: ABORT T_1	ABORT, weil T_1 das geforderte Quorum (4 mal OK) nicht mehr erreichen kann
	C	$UPDATE_{T1}(x)$ OK $UPDATE_{T2}(x)$ WARTE	-	-	-	
	A	$UPDATE_{T2}(x)$ OK $UPDATE_{T3}(x)$ WARTE $UPDATE_{T1}(x)$ PASSIERE	-	-	-	

t	K	Status	Input	Entscheidung	Output	Kommentar
8	A	$UPDATE_{T2}(x)$ OK $UPDATE_{T3}(x)$ WARTE	ABORT T_1	-	-	
	C	$UPDATE_{T2}(x)$ OK	ABORT T_1	-	$A : UPDATE_{T2}(x)$ OK $B : UPDATE_{T2}(x)$ OK $C : UPDATE_{T2}(x)$ OK	
	D		ABORT T_1	-	-	
	E		ABORT T_1	-	-	
	F	$UPDATE_{T3}(x)$ OK	ABORT T_1	-	-	
	G	$UPDATE_{T3}(x)$ OK	ABORT T_1	-	-	
9	D	$UPDATE_{T2}(x)$ OK	$A : UPDATE_{T2}(x)$ OK $B : UPDATE_{T2}(x)$ OK $C : UPDATE_{T2}(x)$ OK	COMMIT T_2	@ALL: COMMIT T_2	
	A	$UPDATE_{T2}(x)$ OK $UPDATE_{T3}(x)$ WARTE	-	-	-	
10	A	$UPDATE_{T3}(x)$ ABGELEHNT	COMMIT T_2	ABORT T_3	@ALL:ABORT T_3	Durch das COMMIT wurden die Zeitstempel aktualisiert. Dadurch sind die Zeitstempel von $F : UPDATE_{T3}(x)$ und $G : UPDATE_{T3}(x)$, die Knoten A in Zeitschritt 5 von Knoten G empfangen hat, veraltet. Deshalb wird ein ABORT ausgelöst.

Aufgabe 2

a)

i	3	14	26	41	56
0 i+1	P11	P26	P41	P51	P61
1 i+2	P11	P26	P41	P51	P61
2 i+4	P11	P26	P41	P51	P61
3 i+8	P11	P26	P41	P51	P3
4 i+16	P26	P41	P51	P61	P11
5 i+32	P41	P51	P61	P11	P26

b) Bei jedem Schritt der Suche kann der Suchbereich weiter eingegrenzt werden, es wird also immer mindestens ein genauerer Eintrag der Fingertabelle als der vorherige Eintrag betrachtet.
Bei einem Kreis mit maximal m verschiedenen möglichen Werten ist der Abstand zwischen dem aktuellen Knoten A_i, an dem im i-ten Schritt gesucht wird, und dem gesuchten Knoten x maximal: $|A_i - x| \leq \left\lfloor \frac{m}{2^i} \right\rfloor$.
Dies ergibt sich daraus, dass der mögliche Bereich immer mindestens halbiert wird, da ansonsten aufeinander folgende Schritte widersprüchliche Ergebnisse liefern.
Beendet werden kann der Algorithmus spätestens, wenn der Abstand zwischen dem aktuellen und dem gesuchten Knoten 1 ist, da dann der nächste Knoten des gesuchte ist.Dann gilt: $|A_i - x| = 1 = \left\lfloor \frac{m}{2^i} \right\rfloor \Leftrightarrow m = 2^i \Leftrightarrow i = log_2 m$
Der Aufwand für die Suche in diese Netzwerk ist somit maximal logarithmisch.

c)
- Suche K90 ($\rightarrow 90 - 64 = 26$) ausgehend von P3: P3 - P26 (gefunden!)
- Suche K8 ausgehend von P11: P11 - P51 - P3 - P11
- Suche K2 ausgehend von P11: P11 - P51 - P61 - P3
- Suche K258 ($\rightarrow 258 - 64 * 4 = 2$) ausgehend von P61: P61 - P3 (gefunden!)

d) Hinzufügen von P33 Suchen der Stelle, wo P33 eingefügt werden muss: vor P41
P41 informiert P33 über seinen Vorgänger und P33 meldet sich bei P26 an.
Dann müssen die anderen Peers über den hinzugefügten Peer informiert werden, damit sie ihre Fingertabellen aktualisieren können. Daraufhin kann P33 die ihm zugeteilten Daten von P41 (von K27 bis K33 (*64)) übernehmen.

e) Fingertabellen nach Hinzufügen von P33:

i	3	11	14	26	41	51	56	61
0 i+1	P11	P14	P26	P33	P51	P56	P61	P3
1 i+2	P11	P14	P26	P33	P51	P56	P61	P3
2 i+4	P11	P26	P26	P33	P51	P56	P61	P3
3 i+8	P11	P26	P26	P41	P51	P61	P3	P11
4 i+16	P26	P33	P33	P51	P61	P3	P11	P14
5 i+32	P41	P51	P51	P61	P11	P26	P26	P33