

Wintersemester 2018/2019

Peer-to-Peer und Cloud Computing

Lösungsvorschläge zu Aufgabenblatt 4

1 Rechenaufgabe zu Symphony (6 Punkte)

Lesen Sie den wissenschaftlichen Beitrag *Symphony: Distributed Hashing in a Small World* (im Digi-campus sowie →hier verfügbar). Beantworten Sie dazu die folgenden Fragen.

Gegeben sei ein Symphony-Ring mit den Knoten $v_{0,03}$, $v_{0,13}$, $v_{0,2}$, $v_{0,27}$, $v_{0,39}$, $v_{0,42}$, $v_{0,47}$, $v_{0,62}$, $v_{0,75}$ und $v_{0,89}$ (siehe Abbildung ??). Es handelt sich um einen Ring mit bidirektionaler Kommunikation.

1. Für welche Schlüssel ist Knoten $v_{0,42}$ zuständig? (1 Punkt)

Lösung

Für alle Daten d mit $0,39 < (f \circ h)(d) \leq 0,42$ (wobei $f \circ h$ die in Symphony verwendete Hashfunktion ist, siehe Lösung zur Aufgabe *Symphony: Grundlegendes*).

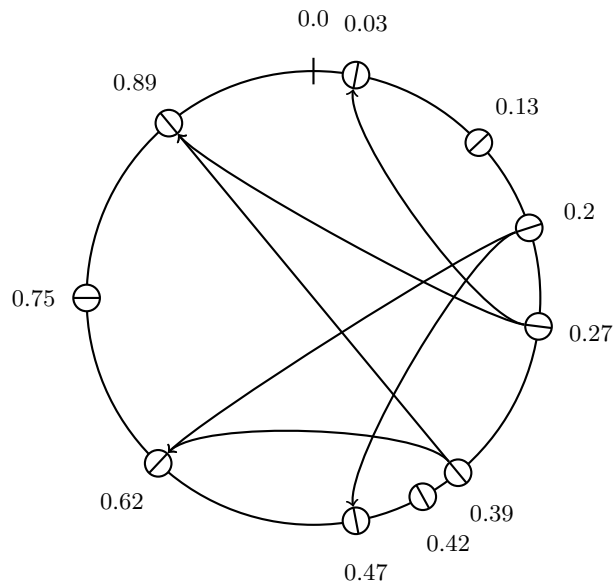
2. Berechnen Sie für den Knoten $v_{0,47}$ den Schätzwert für die Anzahl von Knoten im Netz (basierend auf dem Estimation-Protokoll aus o. g. wissenschaftlichem Beitrag mit $s = 3$). (1 Punkt)

Lösung

$$\tilde{n} = \frac{s}{X_{s_{0,47}}} = \frac{s}{\sum_{i \in \{0,42; 0,47; 0,62\}} l_i} = \frac{3}{0,42 - 0,39 + 0,47 - 0,42 + 0,62 - 0,47} = \frac{3}{0,23} \approx 13,04$$

3. Zeichnen Sie die Long-Distance-Links ($k = 2$) für die Knoten $v_{0,2}$, $v_{0,27}$ und $v_{0,39}$ unter Benutzung der in Abbildung ?? gegebenen „Zufallszahlen“, der in Symphony genutzten *Probability-Distribution-Function* (PDF) in Abbildung ?? ein. (2 Punkte)

Lösung



$e^{\ln n * (rand() - 1, 0)}$	genutzt von Peer	\Rightarrow LDL zu Peer
0,41	0,2	0,62
0,23	0,2	0,47
0,51	0,27	0,89
0,67	0,27	0,03
0,17	0,39	0,62
0,37	0,39	0,89

4. Anschließend fordert Knoten $v_{0,13}$ Daten mit dem Schlüssel 0,88 an. Geben Sie den vollständigen Anfragepfad an und begründen Sie ihn. (2 Punkt)

Lösung

Annahme: Bidirektionaler Ring.

- a) Zuerst sucht sich $v_{0,13}$ aus seiner Liste an Nachbarn denjenigen aus, der die Distanz zum Schlüssel 0,88 minimiert. Hätte er einen Long-Distance-Link zu $v_{0,89}$, würde er die Suche an diesen weitergeben – da aber nichts über die Long-Distance-Links bekannt ist, wird davon ausgegangen, dass dem nicht so ist. Das gleiche gilt für einen möglichen Long-Distance-Link zu $v_{0,75}$. Es müssen also nur die SDLs von $v_{0,13}$ betrachtet werden, für die gilt:

$$\arg \min_{v_k \in \{v_{0,03}; v_{0,2}\}} (\min(|k - 0,88|; 1 - |k - 0,88|)) = v_{0,03}$$

Falls mehrere Nachbarn die Distanz zu $v_{0,88}$ gleich stark minimieren, liefert $\arg \min$ einen der Kandidaten gleichverteilt zufällig aus der Menge. Somit wird die Anfrage an $v_{0,03}$ weitergeleitet.

- b) Egal welche Long-Distance-Links der Knoten $v_{0,03}$ auch unterhielte, die Distanz zu 0,88 wird am meisten durch ein Weiterleiten an den Knoten $v_{0,89}$ verringert.

$$\arg \min_{v_k \in \{v_{0,13}; v_{0,89}\}} (\min(|k - 0,88|; 1 - |k - 0,88|)) = v_{0,89}$$

Auch hier gilt die Definition von $\arg \min$ der letzten Teilaufgabe. Somit wird die Anfrage an $v_{0,89}$ weitergeleitet.

- c) $v_{0,89}$ ist das Ziel der Suche, denn er ist für den Hash-Bereich $(0,75; 0,89]$ zuständig.

2 Rechenaufgabe zu Chord (15 Punkte)

Gegeben sei ein Chord-Ring ($m = 6$, mit Fingern) mit den Knoten: $v_1, v_8, v_9, v_{21}, v_{32}, v_{38}, v_{42}, v_{58}$. Geben Sie Ihre Rechenwege an!

- Erstellen Sie die Routingtabellen für alle Knoten. (4 Punkte)

Lösung

Zusätzlich zu den folgenden Fingertabellen gehören zu den Routingtabellen der Knoten auch die Verbindungen zu ihren jeweiligen Vorgängern und Nachfolgern im Ring.

Knoten v_1			Knoten v_8			Knoten v_9		
		Zieladr.			Zieladr.			Zieladr.
$1 + 2^0 = 2$	\Rightarrow	8	$8 + 2^0 = 9$	\Rightarrow	9	$9 + 2^0 = 10$	\Rightarrow	21
$1 + 2^1 = 3$	\Rightarrow	8	$8 + 2^1 = 10$	\Rightarrow	21	$9 + 2^1 = 11$	\Rightarrow	21
$1 + 2^2 = 5$	\Rightarrow	8	$8 + 2^2 = 12$	\Rightarrow	21	$9 + 2^2 = 13$	\Rightarrow	21
$1 + 2^3 = 9$	\Rightarrow	9	$8 + 2^3 = 16$	\Rightarrow	21	$9 + 2^3 = 17$	\Rightarrow	21
$1 + 2^4 = 17$	\Rightarrow	21	$8 + 2^4 = 24$	\Rightarrow	32	$9 + 2^4 = 25$	\Rightarrow	32
$1 + 2^5 = 33$	\Rightarrow	38	$8 + 2^5 = 40$	\Rightarrow	42	$9 + 2^5 = 41$	\Rightarrow	42

Knoten v_{21}			Knoten v_{32}			Knoten v_{38}		
		Zieladr.			Zieladr.			Zieladr.
$21 + 2^0 = 22$	\Rightarrow	32	$32 + 2^0 = 33$	\Rightarrow	38	$38 + 2^0 = 39$	\Rightarrow	42
$21 + 2^1 = 23$	\Rightarrow	32	$32 + 2^1 = 34$	\Rightarrow	38	$38 + 2^1 = 40$	\Rightarrow	42
$21 + 2^2 = 25$	\Rightarrow	32	$32 + 2^2 = 36$	\Rightarrow	38	$38 + 2^2 = 42$	\Rightarrow	42
$21 + 2^3 = 29$	\Rightarrow	32	$32 + 2^3 = 40$	\Rightarrow	42	$38 + 2^3 = 46$	\Rightarrow	58
$21 + 2^4 = 37$	\Rightarrow	38	$32 + 2^4 = 48$	\Rightarrow	58	$38 + 2^4 = 54$	\Rightarrow	58
$21 + 2^5 = 53$	\Rightarrow	58	$32 + 2^5 = 64$	\Rightarrow	1	$38 + 2^5 = 70$	\Rightarrow	8

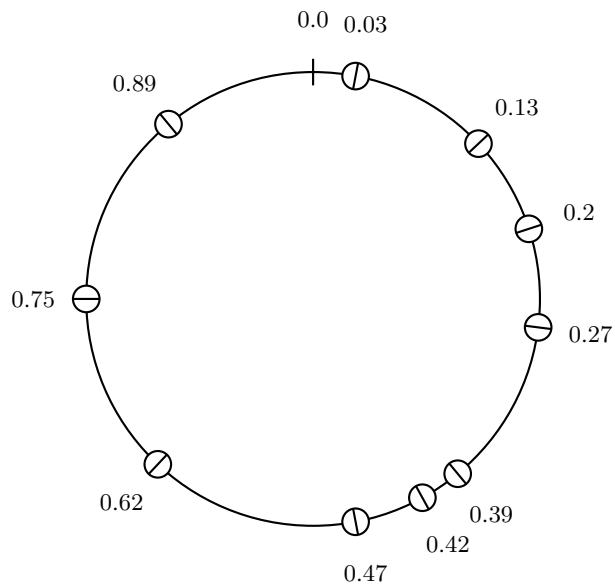


Abbildung 1: Symphony-Netzwerk für Aufgabe ??.

$e^{\ln n * (rand() - 1, 0)}$	genutzt von Peer
0,41	0,2
0,23	0,2
0,51	0,27
0,67	0,27
0,17	0,39
0,37	0,39

Abbildung 2: Zufallszahlen, die in den entsprechenden Schritten von der PDF $e^{\ln n * (rand() - 1, 0)}$ generiert wurden.

Knoten v_{42}		Knoten v_{58}	
	Zieladr.		Zieladr.
$42 + 2^0 = 43$	$\Rightarrow 58$	$58 + 2^0 = 59$	$\Rightarrow 1$
$42 + 2^1 = 44$	$\Rightarrow 58$	$58 + 2^1 = 60$	$\Rightarrow 1$
$42 + 2^2 = 46$	$\Rightarrow 58$	$58 + 2^2 = 62$	$\Rightarrow 1$
$42 + 2^3 = 50$	$\Rightarrow 58$	$58 + 2^3 = 66$	$\Rightarrow 8$
$42 + 2^4 = 58$	$\Rightarrow 58$	$58 + 2^4 = 74$	$\Rightarrow 21$
$42 + 2^5 = 74$	$\Rightarrow 21$	$58 + 2^5 = 90$	$\Rightarrow 32$

2. Ein neuer Knoten mit der ID 41 nimmt Kontakt mit Knoten v_1 auf, um ins Netzwerk aufgenommen zu werden.

- Welche Schritte werden unternommen bis der neue Knoten Teil des Netzwerks ist? (3 Punkte)

Lösung

- Aufnahme-Anfrage an Knoten v_1 .
 - v_1 startet Suche nach Knoten, der für Adresse 41 zuständig ist.
 - v_1 checkt, ob sein Nachfolger, v_8 , für 41 zuständig ist: Nein!
 - v_1 schaut in seiner Fingertabelle nach, findet v_{38} als „Nächstkleineren“. Also Anfrage an v_{38} .
 - v_{38} startet Suche nach Knoten, der für Adresse 41 zuständig ist.
 - v_{38} checkt, ob sein Nachfolger, v_{42} , für 41 zuständig ist: Ja!
 - v_{41} verbindet sich mit v_{42} und v_{38} .
 - v_{41} erstellt seine eigene Routingtabelle (siehe nächste Teilaufgabe).
 - v_{41} benachrichtigt Knoten, die ihre Routingeinträge kontrollieren müssen – eventuell rekursiv (siehe nächste Teilaufgabe).
 - Eventuell: Daten, für die er zuständig ist, werden an v_{41} übertragen.
- Geben Sie die neue Routingtabelle für Knoten v_{41} an. (1 Punkt)

Lösung

Fingertabelle:

	Zieladr.
$41 + 2^0 = 42 \Rightarrow$	42
$41 + 2^1 = 43 \Rightarrow$	58
$41 + 2^2 = 45 \Rightarrow$	58
$41 + 2^3 = 49 \Rightarrow$	58
$41 + 2^4 = 57 \Rightarrow$	58
$41 + 2^5 = 73 \Rightarrow$	9

Außerdem in Routing-Tabelle: v_{38} , v_{42} .

- Welche Knoten müssen von v_{41} dazu aufgefordert werden, ihre Routingtabellen zu aktualisieren? (1 Punkt)

Lösung

Sei $f \equiv \text{findPredecessor}$. Update-Tabelle für v_{41} :

	Zieladr.
$41 - 2^0 = 40 \xrightarrow{f}$	38
$41 - 2^1 = 39 \xrightarrow{f}$	38
$41 - 2^2 = 37 \xrightarrow{f}$	32
$41 - 2^3 = 33 \xrightarrow{f}$	32
$41 - 2^4 = 25 \xrightarrow{f}$	21
$41 - 2^5 = 9 \xrightarrow{f}$	9

- Geben Sie die aktualisierten Routingtabellen der anderen Knoten an. (2 Punkte)

Lösung

v_{42} setzt seinen Vorgänger auf v_{41} , v_{38} setzt seinen Nachfolger auf v_{41} . Der Knoten v_{21} wird zwar zum Kontrollieren seiner Routingtabelle aufgefordert, muss aber keine Änderungen vornehmen. Die neuen Fingertabellen der Knoten v_9 , v_{32} und v_{38} sind:

	Ziel		Ziel		Ziel
$9 + 2^0 = 10 \Rightarrow$	21	$32 + 2^0 = 33 \Rightarrow$	38	$38 + 2^0 = 39 \Rightarrow$	41
$9 + 2^1 = 11 \Rightarrow$	21	$32 + 2^1 = 34 \Rightarrow$	38	$38 + 2^1 = 40 \Rightarrow$	41
$9 + 2^2 = 13 \Rightarrow$	21	$32 + 2^2 = 36 \Rightarrow$	38	$38 + 2^2 = 42 \Rightarrow$	42
$9 + 2^3 = 17 \Rightarrow$	21	$32 + 2^3 = 40 \Rightarrow$	41	$38 + 2^3 = 46 \Rightarrow$	58
$9 + 2^4 = 25 \Rightarrow$	32	$32 + 2^4 = 48 \Rightarrow$	58	$38 + 2^4 = 54 \Rightarrow$	58
$9 + 2^5 = 41 \Rightarrow$	41	$32 + 2^5 = 64 \Rightarrow$	1	$38 + 2^5 = 70 \Rightarrow$	8

Zusätzlich fordert jeder der Knoten seinen Vorgänger zum Kontrollieren seiner Routingtabellen auf, was in diesem Fall zu einem Update der Routingtabelle von v_8 über v_9 führt:

	Zieladr.
$8 + 2^0 = 9 \Rightarrow$	9
$8 + 2^1 = 10 \Rightarrow$	21
$8 + 2^2 = 12 \Rightarrow$	21
$8 + 2^3 = 16 \Rightarrow$	21
$8 + 2^4 = 24 \Rightarrow$	32
$8 + 2^5 = 40 \Rightarrow$	41

3. Anschließend fordert Knoten v_{41} Daten mit dem Schlüssel 9 an.

- Welche Knoten werden von welchen Knoten in welcher Reihenfolge nach den Daten gefragt? (2 Punkte)

Lösung

- v_{41} sucht nach Knoten, der für Adresse 9 zuständig ist.
 - v_{41} checkt, ob sein Nachfolger, v_{42} , für 9 zuständig ist: Nein!
Achtung: Dazu muss er *keine* Nachricht an v_{42} schicken!
 - v_{41} schaut in seiner Fingertabelle nach, findet v_{58} als „Nächstkleineren“ nach $9 = 73 \bmod 64$.
 \Rightarrow Anfrage an v_{58}
 - v_{58} sucht nach Knoten, der für Adresse 9 zuständig ist.
 - v_{58} checkt, ob sein Nachfolger, v_1 , für 9 zuständig ist: Nein!
 - v_{58} schaut in seiner Fingertabelle nach, findet v_8 .
 \Rightarrow Anfrage an v_8
 - v_8 sucht nach Knoten, der für Adresse 9 zuständig ist.
 - v_8 checkt, ob sein Nachfolger, v_9 , für 9 zuständig ist: Ja!
 - v_8 liefert Kontaktdaten von v_9 an v_{41} zurück.
- Welcher Knoten liefert schließlich das Ergebnis der Suche an v_{41} zurück? (1 Punkt)

Lösung

Am Ende liefert v_8 die Adresse von v_9 an v_{41} zurück.

4. Knoten v_{21} fällt aus. Welche Knoten aktualisieren nun unmittelbar welche Informationen? (1 Punkt)

Lösung

- Knoten bemerken über Keep-Alive-Nachrichten den Ausfall von v_{21}

- v_{32} löscht Verbindung zu v_{21} (seinem Vorgänger)
- v_9 löscht Verbindung zu v_{21} (seinem Nachfolger), setzt v_{32} als neuen Nachfolger
 \Rightarrow Ringstruktur erhalten
- nicht unmittelbar, aber periodisch: Stabilisation-Protokoll repariert den Rest

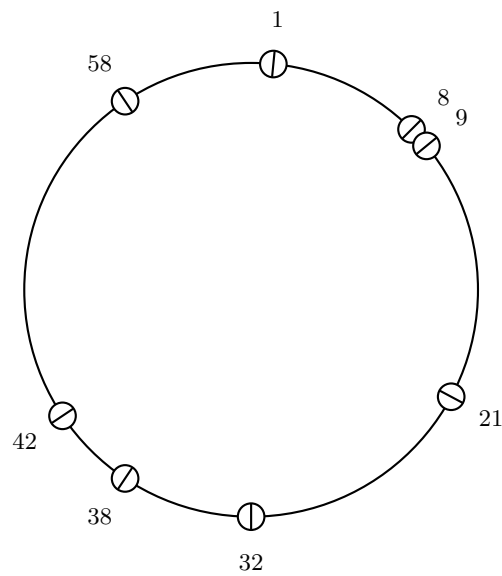


Abbildung 3: Chord-Netzwerk zu Aufgabe ??.