Institut für Informatik

Lehrstuhl für Technische Informatik
Prof. Dr. Björn Scheuermann

Peer-to-Peer-Systeme WS 2015/16

Übungsblatt 3b

Besprechung am 3. Dezember 2015

Aufgabe 12

Stellen wir uns eine DHT vor, die auf folgendem Grundprinzip beruht: Die n Peers werden von 0 bis $n-1$ durchnummeriert, jeder Peer kennt seine ID. Die Schlüssel sind natürliche Zahlen. Als Hashfunktion wird

$$h : \begin{cases} \mathbb{N} \to \{0, \dots, n-1\} \\ k \mapsto k \bmod n \end{cases}$$

verwendet. Der Peer mit ID i verwaltet alle Schlüssel k, für die $h(k) = i$.

Warum ist dies keine gute Grundlage für eine DHT? Nenne mindestens zwei schwerwiegende Probleme.

Lösung:

- Beim Hinzukommen oder beim Wegfall eines Peers ändert sich die Hashfunktion so, dass die Zuordnung der Schlüssel zu den verbleibenden Peers vollständig anders aussehen kann. Bei jeder Änderung der Peer-Menge sind deswegen umfangreiche Umstrukturierungen im gesamten Netzwerk die Folge.
- Die (verteilte, dezentrale) Zuordnung eindeutiger (und hintereinanderliegender) Nummern für alle Peers ist nicht ohne weiteres möglich.

Aufgabe 13

Angenommen, ich lasse mich von deinen guten Argumenten gegen das Konstruktionsprinzip in der vorangegangenen Aufgabe nicht aufhalten und baue eine DHT mit der dort beschriebenen Zuordnung von Schlüsseln zu Peers. Ich schlage zwei Möglichkeiten vor, das Overlay-Netzwerk zu konstruieren:

In Variante 1 werden die Peers nach ihren IDs sortiert in einem Kreis angeordnet und jeder Peer wird mit seinen zwei Nachbarn verbunden, Peer i also mit $i - 1$ und mit $i + 1$ (jeweils modulo n). Geroutet wird durch Weiterreichen an den Nachbarn, der näher am gesuchten Schlüssel liegt.

In Variante 2 werden alle Peers mit dem Peer mit der ID 0 verbunden. Sonst existieren keine Verbindungen. Geroutet wird, indem eine Nachricht zunächst an Peer 0 übergeben und dann von diesem an den Ziel-Peer weitergeleitet wird.

(a) Was ist die Routing-Komplexität dieser beiden Varianten, also (asymptotisch) die durchschnittliche Zahl von Hops zum Ziel?

> **Lösung:**
> Variante 1: $O(n)$, da typischerweise um 1/4 des Kreises geroutet werden muss.
> Variante 2: $O(1)$, da maximal zwei Hops immer ausreichen.

(b) Was ist die Worst-Case-Zustandskomplexität, also die (asymptotische) Anzahl an anderen Peers, mit denen jeder Peer eine Verbindung unterhalten muss?

> **Lösung:**
> Variante 1: $O(1)$, da jeder Peer genau zwei andere kennt.
> Variante 2: $O(n)$, da Peer 0 alle anderen Peers kennen muss.

Aufgabe 14

Betrachte die DHT CAN über dem Schlüsselraum $[0,1)^2$. Ignoriere den „Wraparound“, also die Tatsache, dass das Gebiet als Torus und nicht als Quadrat behandelt wird. Nimm außerdem an, dass die Zahl der Peers n eine Viererpotenz ist, also $n = 4^i$ für ein $i \in \mathbb{N}$, und dass die Gebietsaufteilung „perfekt“ ist, dass also alle Peers gleich große Gebiete haben.

(a) Wie groß ist der Netzwerkdurchmesser, also die maximale Distanz in Hops zwischen zwei Peers, abhängig von der Zahl der Peers n?

> **Lösung:** Das Gebiet wird in $\sqrt{n} \times \sqrt{n}$ kleine Quadrate aufgeteilt. Der längste Weg ist der von einer Ecke in die gegenüberliegende. Dafür müssen $\sqrt{n}-1$ „horizontale“ und $\sqrt{n}-1$ „vertikale“ Schritte gemacht werden. Also ist der Durchmesser $2(\sqrt{n}-1)$.

(b) Wenn du jetzt eine Verallgemeinerung auf d Dimensionen betrachtest (und entsprechend annimmst, dass die Zahl der Peers eine ganzzahlige Potenz von 2^d ist), wie groß ist dann der Netzwerkdurchmesser, in Abhängigkeit von d und n?

> **Lösung:** Analog zur vorherigen Teilaufgabe ist die „perfekte“ Aufteilung des d-dimensionalen Gebietes die Unterteilung in n kleine, d-dimensionale Hyperwürfel. Entlang jeder Dimension liegen $\sqrt[d]{n}$ solche Hyperwürfel, insgesamt sind also (wieder von einer Ecke in die gegenüberliegende) maximal $\sqrt[d]{n}-1$ Schritte in jeder Dimension und damit insgesamt $d(\sqrt[d]{n}-1)$ Schritte notwendig.

(c) Was ändert sich, wenn du jetzt den Wraparound noch in Betracht ziehst?

> **Lösung:** Der längste Weg führt jetzt von einer Ecke in ein Gebiet direkt hinter der Mitte des Hash-Wertebereichs. Dafür sind $\frac{1}{2}\sqrt[d]{n}$ Schritte in jeder Dimension nötig, also insgesamt $\frac{d}{2}\sqrt[d]{n}$ Schritte.
> Asymptotisch entspricht das übrigens genau dem Average-Case-Resultat, das für allgemeine CAN-Räume in der Vorlesung genannt wurde: $O(dn^{1/d})$