Institut für Informatik

Lehrstuhl für Technische Informatik
Prof. Dr. Björn Scheuermann

Peer-to-Peer-Systeme WS 2015/16

Übungsblatt 2

Besprechung am 12. November 2015

Aufgabe 6

Stellen wir uns ein hierarchisches Peer-to-Peer-Overlay mit *drei* Hierarchiestufen vor: Es gibt „normale" Peers, Superpeers und Super-Duper-Peers.

(a) Nehmen wir an, dass ein Superpeer 100 normale Peers betreut und ein Super-Duper-Peer für 100 Superpeers zuständig ist. Wie viele Super-Duper-Peers wären in einem Netzwerk mit 999 000 Peers notwendig?

> **Lösung:** Ein Super-Duper-Peer verwaltet (indirekt) $100 \cdot 100 = 10\,000$ normale Peers plus 100 Superpeers plus sich selbst, insgesamt also $10\,101$ Peers. Es sind damit $\lceil 999\,000/10\,101 \rceil = 99$ Super-Duper-Peers notwendig.

(b) Überlege, wie in einem solchen Netzwerk die Bearbeitung von Suchanfragen erfolgen könnte. Welche Informationen sollte ein Superpeer speichern? Was sollte ein Super-Duper-Peer wissen? Wie würde eine Suche sinnvollerweise ablaufen?

> **Lösung:** Ein Superpeer sollte wissen, welcher seiner angeschlossenen Peers welche Datei(en) vorhält.
>
> Ein Super-Duper-Peer sollte wissen, welche Dateien über welchen angeschlossenen Superpeer (indirekt) erreicht werden können; er muss *nicht* wissen, welche(r) Peer(s) unterhalb dieses Superpeers eine bestimmte Datei besitzen.
>
> Ein möglicher Ansatz für ein Suchverfahren wäre der folgende: Ein Peer würde seine Suchanfrage zunächst an seinen Superpeer richten. Fall sie lokal beantwortet werden kann, wird sie nicht weitergegeben. Falls keine oder nicht genug Treffer bei den Peers am selben Superpeer zusammenkommen, gibt der Superpeer die Frage an seinen Super-Duper-Peer weiter. Falls die Suche auch innerhalb der Gruppe der Superpeers dieses Super-Duper-Peers nicht beantwortet werden kann, wird sie im Netz der Super-Duper-Peers geflutet.

Aufgabe 7

Betrachte ein hypothetisches hierarchisches Overlay ähnlich Gnutella 0.6. Wir nehmen an, dass darin jeder Superpeer mit genau fünf anderen Superpeers verbunden ist.

Nun setzt einer der Superpeers eine Suchanfrage im Superpeer-Overlay ab. Dabei setzt er Dynamic Querying ein. Ein erster Probe-Suchlauf über 10 Superpeers hat x Treffer geliefert ($1 \leq x < 150$).

Mache einen Vorschlag für die Wahl der TTL bei einer anschließenden „vollen" Suche (abhängig von x), wenn diese neue Suche 150 Treffer erzielen soll.

Lösung: Die getesteten Superpeers haben jeder durchschnittlich $x/10$ Treffer gebracht. Wenn dies generell gilt, sollten bei der Suche $150 \cdot 10/x$ Superpeers erreicht werden.

Wir wissen (aus einer früheren Übungsaufgabe), dass wir in einem solchen Netz mit einer TTL von t (höchstens)

$$5 \cdot \frac{4^t - 1}{3}$$

Peers erreichen werden. Außerdem haben wir diskutiert, dass die tatsächlich erreichte Zahl an Peers – jedenfalls vor Beginn der Annäherung an die Gesamtgröße des Netzes – nicht wesentlich kleiner ist (denn sonst wäre der typische Durchmesser eines solchen Graphen nicht logarithmisch in der Zahl der Peers!). Wir können diese Formel also als brauchbare Näherung für die Zahl der erreichten Peers bei hinreichend kleiner TTL verwenden.

Deshalb sollte folgende Bedingung erfüllt sein:

$$\begin{aligned} & 5 \cdot \frac{4^t - 1}{3} \geq 150 \cdot \frac{10}{x} \\ \Rightarrow \quad & 4^t \geq \frac{900}{x} + 1 \\ \Rightarrow \quad & t \geq \log_4 \left(\frac{900}{x} + 1 \right). \end{aligned}$$

Da t eine Ganzzahl sein muss, sollte man also

$$t = \left\lceil \log_4 \left(\frac{900}{x} + 1 \right) \right\rceil$$

wählen.

Aufgabe 8

Schreibe ein Programm (in einer Programmiersprache deiner Wahl), das für vorgegebene Parameter n, p Gilbert-Zufallsgraphen erzeugt.

Das Programm soll dann die durchschnittliche Pfadlänge des erzeugten Graphen bestimmen und ausgeben. Damit für die Bestimmung der durchschnittlichen Pfadlänge nicht alle Knotenpaare untersucht werden müssen, genügt es, wenn das Programm 100 Knotenpaare zufällig auswählt und diese Paare als Stichprobe für die Berechnung der Durchschnittspfadlänge verwendet.

Außerdem soll für die Graphen der sogenannte Clustering-Koeffizient bestimmt werden. Der Clustering-Koeffizient eines Graphen ist ein Maß für die Lokalität des Graphen, also wie „dicht" die Umgebung eines Knotens typischerweise vernetzt ist. Um den Clustering-Koeffizienten eines Graphen zu bestimmen, berechnet man zunächst den Clustering-Koeffizienten seiner einzelner Knoten. Dafür betrachtet man alle Paare von Nachbarn eines gegebenen Knotens. Für jedes dieser Paare bestimmt man, ob die beiden Knoten selbst auch wieder durch eine Kante verbunden sind. Angenommen, der Knoten hat M mögliche Paare von Nachbarn, von denen m durch eine Kante verbunden sind. Dann ist der Clustering-Koeffizient des Knotens m/M. Der Clustering-Koeffizient eines Graphen ist der durchschnittliche Clustering-Koeffizient seiner Knoten; Knoten mit Grad kleiner 2 bleiben unberücksichtigt.

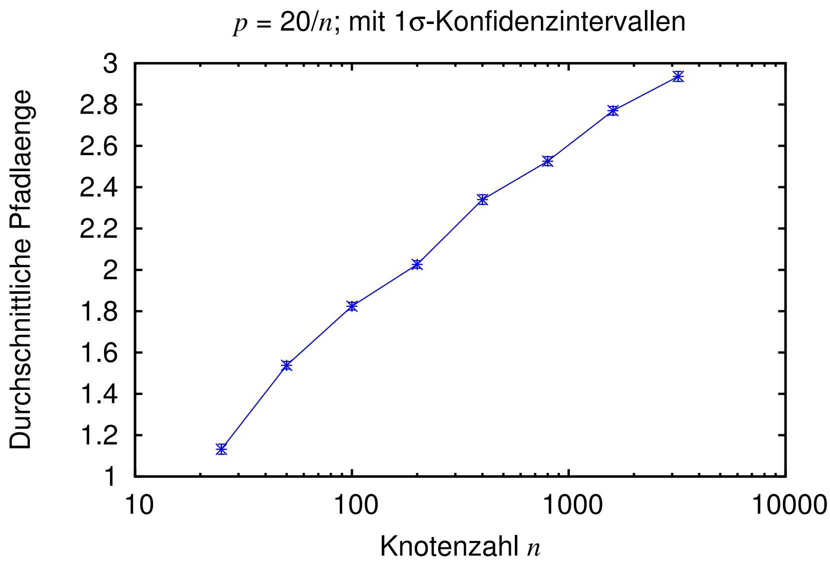
Nutze dein Programm und beantworte die folgenden Fragen:

(a) Eine bekannte Aussage über Zufallsgraphen ist, dass die durchschnittlichen Pfadlängen nur sehr langsam mit der Knotenzahl wachsen. Die Theorie sagt: Bei konstantem durchschnittlichem Knotengrad wächst die durchschnittliche Pfadlänge logarithmisch mit der Knotenzahl. Wie musst Du p abhängig von n wählen, damit der Knotengrad bei steigender Größe des Graphen konstant bleibt?

> **Lösung:** Der Erwartungswert für die Zahl der Nachbarn eines Knotens in einem Gilbert-Graphen mit Parametern n, p ist
>
> $$p \cdot (n-1) \approx pn.$$
>
> Wenn dieser Erwartungswert bei wachsendem n konstant gehalten werden soll, muss p proportional zu $1/n$ fallen.
> p sollte also folgendermaßen gewählt werden:
>
> $$p = \frac{\beta}{n}$$
>
> mit einer geeigneten Konstanten β.

(b) Erstelle ein Diagramm, das die Abhängigkeit der durchschnittlichen Pfadlänge von der Knotenzahl zeigt und dabei den durchschnittlichen Knotengrad konstant hält. Wähle dafür geeignete Parameter(bereiche). Kannst Du in dem Diagramm das von der Theorie vorhergesagte logarithmische Wachstum nachweisen? (Hinweis: Für die visuelle Darstellung eines logarithmischen Wachstums lohnt es sich, über geeignete Achsenskalierungen im Diagramm nachzudenken!)

Lösung:



(c) Untersuche mithilfe geeigneter Darstellung(en), wie der Clustering-Koeffizient von n und p abhängt. Kannst Du das beobachtete Verhalten erklären?

Lösung: Der erwartete Clustering-Koeffizient ist unabhängig von n und gleich p (Begründung: s. Vorlesung).