

Praktische Themen für die Übung zur Vorlesung „Peer-to-Peer-Systeme und Sicherheit“

Um die Zahl der zu verrechneten Übungsstunden zu erhöhen, können Studenten, die 5 ECTS brauchen, praktische Übungen zusätzlich machen. Dazu wird in einem Bereich eine praktische Aufgabe vorgegeben, welche in kleinere Unteraufgaben unterteilt wird.

Diese Aufgaben sollen bis zum Ende des Semester gelöst und die Ergebnisse am Ende kurz vorgeführt werden. Die Abgabe beinhaltet zusätzlich den Code sowie ein kurzer Bericht mit Dokumentation.

Thema 1: Größenbestimmung in P2P-Netzwerken

In einem beliebigen P2P-Netz (Empfehlung Entangled (Kademlia) in Python oder Freepastry in Java) soll mit versch. Mitteln die Größe abgeschätzt werden. Dies kann z.B. über die Fingertabelle geschehen, wo sich über Knotenanzahl in der Nachbarschaft Hochrechnungen durchführen lassen. Peers könnten auch aktiv zählen und die Teilergebnisse zu einem Gesamtergebnis aggregieren.

Teilaufgabe 1 (bis 19.6.)

Machen Sie sich mit der Implementierung vertraut und realisieren Sie das Ziel auf einfachste Art und Weise.

- a) Beschreiben Sie die wichtigsten Funktionen für die KBR- und DHT-Funktionalität.
- b) Implementieren Sie eine einfache Zählfunktion, die beispielsweise über Nachfolger im Adressraum eine Zählnachricht über alle Peers schickt.

Bei 3 Bearbeitern: beschreiben Sie auch Maintenance-Funtionen der Implementierung

Teilaufgabe 2 (bis 3.7.)

Neben der echten Zählung der Peers, kann auch eine Schätzung über die Größe der Nachbarschaft erfolgen.

- a) Beschreiben Sie einen Weg, wie Sie die Anzahl Hochrechnen können.
- b) Implementierung und Test der Methode

Bei 3 Bearbeitern: eine zweite Variante hochzurechnen.

Teilaufgabe 3 (bis 16.7.)

Optimieren Sie nun die Methoden, indem sie entweder den Zählvorgang parallelisieren oder Zählen und Schätzen kombinieren.

- a) Implementieren
- b) Tests machen und alle Zähl/Schätzfunktionen vergleichen.

Bei 3 Bearbeitern: beides machen.

Demonstration: mehrere Peers auf einem Rechner, die Funktionen zählen bzw. schätzen.

Praktische Themen für die Übung zur Vorlesung „Peer-to-Peer-Systeme und Sicherheit“

Um die Zahl der zu verrechneten Übungsstunden zu erhöhen, können Studenten, die 5 ECTS brauchen, praktische Übungen zusätzlich machen. Dazu wird in einem Bereich eine praktische Aufgabe vorgegeben, welche in kleinere Unteraufgaben unterteilt wird.

Diese Aufgaben sollen bis zum Ende des Semester gelöst und die Ergebnisse am Ende kurz vorgeführt werden. Die Abgabe beinhaltet zusätzlich den Code sowie ein kurzer Bericht mit Dokumentation.

Thema 2: Staukontrolle bei extremen Multipfadsenden

In dieser Aufgabe soll ein P2P-Netzwerk untersucht werden, dass vollvermascht ist, aber jede Nachricht über einen anderen Weg und mehrere Hops zum Ziel schickt. Es gibt dabei einen entfernten Bezug zu Anonymisierung, aber dieser steht hier nicht im Vordergrund. Zur Untersuchung soll das Netz rundenbasiert simuliert werden. Jede Runde werden alle Nachrichten einen Hop weitergeleitet. Da ein Peer nicht beliebig viele Nachrichten empfangen und versenden kann, werden überzählige verworfen.

Die Programmiersprache ist freigestellt. Empfohlen wird eine höhere Sprache wie Java oder Python.

Teilaufgabe 1 (bis 19.6.)

Im ersten Teil soll die Basisfunktionalität der Simulation erstellt werden. Dazu muss eine beliebige Anzahl an Peers erzeugt werden können. Pro Runde sollen die Peers in der Lage sein, eine Nachricht mit einer Anzahl an Hops erstellen zu können und Nachrichten sollen pro Runde einen Hop weiter kommen.

Bei 3 Bearbeitern: erzeugen sie Nachrichten statistisch mit einer konfigurierbaren Last.

Teilaufgabe 2 (bis 3.7.)

In dieser Teilaufgabe soll ressourcenbedingter Verlust hinzugefügt werden. Peers sollen eine konfigurierbare Bandbreite an Nachrichten verarbeiten können. Kommen mehr an, wird zufällig verworfen. Der Sender erhält die Verlust- bzw. Erfolgsmeldung nach 3x Anzahl der Nachrichtenhops Runden.

Bei 3 Bearbeitern: erzeugen sie Nachrichtenströme zwischen Knoten statt Zufallszielen.

Teilaufgabe 3 (bis 16.7.)

Sender sollen nun saturiert sein und so viel senden wollen, wie sie rausschicken können. Die Staukontrolle soll dabei pro Knoten erfolgen. Am Anfang ist der Wert für jeden Knoten 1, d.h. er darf einmal pro Runde in einem Pfad einer Nachricht verwendet werden. Bei erfolgreicher Nachricht erhöht sich der Stauwert für alle Knoten um $\frac{k}{\text{Stauwert}_{\text{Knoten}}}$, wobei k eine

Skalierungsfaktor ist. Im Falle eines Verlusts wird der Stauwert der Knoten des Pfads halbiert. Ermitteln Sie die Verluste und beschreiben Sie, ob und wie sich die Last bei den Knoten auf deren Bandbreite einstellt.

Bei 3 Bearbeitern: statt saturiert und zufallsnachrichten, wieder mit Nachrichtenströmen

Demonstration: Simulation rechnen lassen und Ergebnisse vorstellen.