

Übungsblatt 8

Cassandra

1 Consistent Hashing

Implementieren Sie folgenden vereinfachten Fall für *Consistent Hashing*, das dem Hash-Ring von Cassandra zugrunde liegt. Sie finden ein Code-Gerüst in `bdt-uebung-8-consistent-hashing.zip`.

- Sie können von einer statischen Menge von Knoten ausgehen, d.h. die Knotenmenge steht am Anfang fest, und es können weder Knoten ausfallen noch hinzukommen.
- Sie brauchen keine virtuellen Knoten zu implementieren.
- Sie brauchen keine Replikation von Daten zu implementieren, jeder Datensatz landet auf genau einem Knoten.
- Daten bestehen jeweils nur aus einem Schlüssel vom Typ String, die eigentlichen Nutzdaten brauchen Sie nicht zu abzubilden.
- In Java gibt es ein Interface `NavigableMap` mit einer Implementierung `TreeMap`, das nicht nur die exakte Suche nach Schlüsseln anbietet (`get(key)`), sondern auch Funktionen wie "nächster Eintrag hinter `<key>`" (`ceilingEntry(key)`) und "voriger Eintrag vor `<key>`" (`floorEntry(key)`).
- Im Beispielcode finden Sie ein Interface und eine abstrakte Basisklasse für den Hash-Ring. Außerdem ist eine Hashfunktion in `HashUtils` implementiert, die Sie benutzen können.

2 Consistent Hashing (Fortsetzung)

1. Verteilen Sie eine größere Anzahl (z. B. 1 Million) Datensätze auf 10 Knoten und zählen Sie aus, wie sich die Daten über die Knoten verteilen.
2. Implementieren Sie virtuelle Knoten und prüfen Sie, ob die Verteilung besser wird (d.h. einer Gleichverteilung ähnlicher). Sie können dies am einfachsten umsetzen, indem Sie z. B. statt eines Knotens "host1" mehrere Knoten "host1-1", "host1-2", ..., "host1-100" erzeugen und beim Auszählen diese Suffixe "-1" usw. wieder abschneiden.