

HANDBUCH DEMONSTRATOR ZUR EVALUIERUNG EINES SPEICHERPLATZAGNOSTISCHEN KADEMLIA- PROTOKOLLS

Version 1.0 vom 12.07.2018
Bernd Prünster – bernd.pruenster@a-sit.at

Inhaltsverzeichnis

| | |
|----------------------------------|---|
| Inhaltsverzeichnis | 1 |
| 1. Einleitung | 1 |
| 2. Systemvoraussetzungen | 1 |
| 3. Installation | 2 |
| 3.1. Kompilieren von Source Code | 2 |
| 3.2. Bereitgestelltes Paket | 2 |
| 4. Inbetriebnahme | 2 |
| 4.1. Konfiguration | 2 |
| 4.2. Starten | 2 |
| 4.3. Testen | 2 |
| 5. Handhabung | 3 |
| 6. Lizenz | 3 |
| 7. Verweise | 3 |

1. Einleitung

Der vorliegende Demonstrator versteht sich als Framework, um die Umsetzbarkeit und Praxistauglichkeit eines speicherplatzagnostischen Kademia-Protokolls zu evaluieren. Es handelt sich dabei um eine Weiterentwicklung des *Kademia*-Protokolls [1], das es Knoten ermöglicht durch die Deklaration virtueller Knoten nach Belieben öfter oder seltener für die Speicherung von Daten ausgewählt zu werden. Darüber hinaus werden selbstzertifizierende Identifikatoren im Sinne von *S/Kademia* [2] unterstützt. Als Basis wurde eine bestehende Implementierung¹ herangezogen. Das bereitgestellte Paket wurde als Bibliothek.

Die Zielgruppe sind (Java-)Entwickler im Peer-to-Peer-Bereich, welche die Praxistauglichkeit (insbesondere den Overhead) des Konzepts evaluieren möchten. Mangels integrierter NAT-Traversal-Maßnahmen, wie *Interactive Connectivity Establishment* (ICE) ist ein produktiver Einsatz nur bedingt möglich.

2. Systemvoraussetzungen

Das Framework setzt Java in der Version 1.8 voraus.

Ein Java Development Kit (JDK) in Version 1.8, sowie Apache Maven 3.5 werden benötigt, um den Source-Code zu kompilieren, bzw. die enthaltenen Tests automatisiert auszuführen.

¹ <https://github.com/JoshuaKissoon/Kademia>

Beim erstmaligen Kompilieren wird eine Internetverbindung vorausgesetzt.

Abgesehen davon gibt es keine Einschränkungen, bzw. Systemvoraussetzungen. Das Programm ist sowohl unter Microsoft Windows, Apple macOS, als auch unter Linux lauffähig.

3. Installation

Beim bereitgestellten Paket handelt es sich um ein Framework, das als Bibliothek in andere Programme eingebunden werden kann, daher ist keine Installation vorgesehen. Die enthaltenen Tests, welche die Funktionalität demonstrieren, können automatisiert ausgeführt werden (siehe Abschnitt 4).

3.1. Kompilieren von Source Code

Die Kompilation erfolgt über Apache Maven durch ausführen von `mvn compile`.

3.2. Bereitgestelltes Paket

Der bereitgestellte Quellcode ist zu Teilen in Java und in Kotlin geschrieben. Dies hat keinen Einfluss auf die Kompatibilität zu bestehenden Java-Anwendungen. Eine Integration in andere Projekte ist als Maven-Abhängigkeit möglich.

Die Grundfunktionalität umfasst ein Kademlia-artiges Peer-to-Peer-Netzwerk auf UDP-Basis. Die Besonderheit liegt in der Verwendung von virtuellen Knoten, deren Anzahl jeder Knoten deklariert. Die tatsächlichen Identifikatoren werden mittels der SHA3-Hashfunktion abgeleitet.

Um die bereitgestellte Funktionalität zu demonstrieren, ist eine Test-Suite enthalten. Der entsprechende Code (der auch als Basis für Integration, bzw. Weiterentwicklung herangezogen werden kann) findet sich unter `kademlia/src/test/java/`.

4. Inbetriebnahme

Da es sich um eine Softwarebibliothek handelt, ist eine eigenständige Inbetriebnahme lediglich im Rahmen einer Test-Suite vorgesehen (siehe Abschnitt 4.3).

4.1. Konfiguration

Das bereitgestellte Paket verfügt über keinerlei Konfigurationsparameter.

4.2. Starten

Siehe nächster Abschnitt.

4.3. Testen

Das Programmpaket verfügt über eine integrierte Test-Suite auf Basis von JUnit. Diese umfasst einen Funktionstest, welcher die Funktionalität des Routingprotokolls überprüft. Dazu werden drei Knoten miteinander verbunden, wobei jeder eine unterschiedliche Anzahl virtueller Knoten bereitstellt. Anschließend wird für tausend Datensätze der zuständige Knoten gesucht. Die Auswahl erfolgt auf Basis der Identifikatoren der virtuellen Knoten. Abschließend wird protokolliert, welcher Knoten wie oft ausgewählt.

Die Test-Suite wird durch den Aufruf von `mvn test` gestartet.

5. Handhabung

Da als eigenständig lauffähige Funktionalität lediglich eine Test-Suite bereitgestellt wird, gibt es keine Besonderheiten bei der Handhabung zu beachten (siehe Abschnitt 4.3).

6. Lizenz

Das bereitgestellte Projekt wird unter den Bedingungen der Open-Source-Lizenz für die Europäische Union (EURL) V1.1 bereitgestellt (siehe Lizenz.pdf) und verwendet Bibliotheken der Stiftung SIC (siehe SIC_LICENSE.txt).

7. Verweise

- [1] P. Maymounkov und D. Mazières, „Kademlia: A Peer-to-Peer Information System Based on the XOR Metric,“ in *Peer-to-Peer Systems*, Springer, 2002, pp. 53-65.
- [2] I. Baumgart und S. Mies, „S/Kademlia: A practicable approach towards secure key-based routing,“ in *2007 International Conference on Parallel and Distributed Systems*, 2007.