Message Broker

Seminar Concurrent Programming in C

an der ZHAW am Standort Zürich

Reto Hablützel

April 27, 2014

1 Einleitung

Ein Message Broker ist eine zentrale Kommunikatinsschnittstelle für mehrere Programme. Sie ermöglicht es, dass verschiedenste Programme miteinander kommunizieren können, indem Nachrichten aneinander versendet werden. Ein Message Broker bietet typischerweise verschiedene Protokolle zur Kommunikation an.

stomp erwähnen

2 Funktionalität

Dieses Kapitel erläutert die Funktionalitäten, die im Message Broker im Rahmen dieses Seminars umgesetzt wurden.

2.1 Verbindungsauf- und abbau

Jede Verbindung zwischen einem Client und dem Message Broker beginnt mit einem kurzen Handshake. Der Client teilt dem Message Broker seinen Namen mit und der Message Broker bestätigt den Empfang. Sobald der Client die Kommunikation beenden will, teilt er dies dem Message Broker mit einer Disconnect Nachricht mit und kriegt wiederum eine Bestätigung.

2.2 Nachrichten empfangen

Ein verbundener Client kann einen Topic abonnieren und kriegt so vom Message Broker sämtliche Nachrichten zu diesem Topic zugestellt. Topics werden dynamisch erstellt, sobald sicher der erste Client dafür interessiert. Ein Client, der Nachrichten von einem Topic abonniert hat, wird Subscriber genannt.

2.3 Nachrichten senden

Ein verbundener Client kann Nachrichten zu einem bestimmten Topic an den Message Broker senden. Ein solcher Client wird als Publisher bezeichnet.

3 Konzept

3.1 Protokoll

Für die Umsetzung wurde eine abgespeckte Version vom STOMP[1] Protokoll implementiert. Die implementierten Kommandos werden unter aufgelistet. Auf sämtliche Kommandos vom Client kann der Server mit ERROR antworten, falls etwas schief gelaufen ist. Jedes Kommando hat die folgende Struktur:

Auf der ersten Zeile steht der Name des Kommandos in Grossbuchstaben. Auf folgenden Zeilen stehen durch Doppelpunkt separierte Key-Value Paare,

[1] https://stomp.github.io/

welche Header Werte darstellen. Nach den Header Werten folgt der Content, welcher durch eine Leerzeile vom Header getrennt ist. Das Ende eines Kommandos markiert wiederum eine Leerzeile gefolgt vom Null-Byte (hier ^@):

Listing 1: Struktur eines Kommandos

```
COMMAND
key: value
key: value
...

Content

^@
```

3.1.1 CONNECT

Wird vom Client an den Message Broker zum Beginn der Verbindung gesendet. Hat einen zwingenden Header 'login', welcher den Client identifiziert. Hat keinen Content. Sobald der Message Broker das Kommando den Client aufgenommen hat, sendet er zur Bestätigung CONNECTED.

3.1.2 CONNECTED

Wird vom Message Broker als Antwort auf das CONNECT Kommando versendet falls die Verbindung erfolgreich erstellt werden konnte. Hat weder Header noch Content.

3.1.3 ERROR

Wird vom Message Broker in verschiedenen Szenarien an den Client gesendet um einen Fehler mitzuteilen. Hat einen zwingenden Header 'message', der die Fehlermeldung enthält und keinen Content.

3.1.4 SEND

Wird von einem Client an den Message Broker gesendet um einen Nachricht zuzustellen. Hat einen zwingenden Header 'topic', der den Topic identifiziert, an den die Nachricht gesendet wird. Im Content steht der eigentliche Inhalt der Nachricht.

3.1.5 SUBSCRIBE

Wird von einem Client an den Message Broker gesendet um einen Topic zu abonnieren. Hat einen zwingenden Header 'destination', welcher den Topic identifiziert. Hat keinen Content.

3.1.6 MESSAGE

Wird vom Message Broker an den Client gesendet um eine Nachricht zuzustellen. Hat einen zwingenden Header 'destination', der den Topic identifiziert. Die Nachricht ist im Content.

3.1.7 DISCONNECT

Wird vom Client an den Message Broker gesendet um die Verbindung zu beenden. Hat weder Header noch Content.

3.1.8 RECEIPT

Wird vom Message Broker an den Client gesendet, sobald der Message Broker den DISCONNECT akzeptiert hat. Hat weder Header noch Content.

3.2 Speicherstruktur

Auf dem Server existieren zwei globale Listen: Die eine enthält alle Topics verknüpft mit einer Liste von Subscribern zum jeweiligen Topic. Die zweite Liste enthält alle Messages, die vom Message Broker empfangen wurden. Zudem enthält jede Nachricht eine Liste von Statistiken. Eine Statstik ist eine Tripel aus Subscriber, Anzahl versuchter Zustellungen und einen Timestamp der letzten erfolglosen Zustellung. Diese wird verwendet um zu bestimmen, welche Subscriber die Nachricht erhalten haben und ob ein erneuter Versuch der Zustellung gemacht werden soll.

Listing 2: Speicherstruktur

```
Topics:
  - topic1
      - subscriber1
      - subscriber2
  - topic2
      - subscriber1
      - subscriber3
       ..
   ..

Messages:
  - message1
      - topic
      - content
      - statistic1
          - last fail
          - nattempts
          - subscriber1
      - statistic2
```

```
        - last fail
        - nattempts
        - subscriber2
- message2
    ..
```

3.2.1 Einfügen einer neuen Nachricht

Wenn ein Client eine Nachricht an den Message Broker sendet, kopiert er die Liste von Subscribern von dem Topic, an den die Nachricht versendet wurde. Dann wird ein neuer Eintrag in der Liste der Nachrichten erstellt und die Subscriber werden in eine Liste von Statistiken überführt.

3.3 Komponenten

3.3.1 Broker

Pro verbundenem Client existiert ein Thread, der den Broker repräsentiert. Der Broker hält eine TCP Verbindung aufrecht und ist stets bereit, Kommandos zu lesen. Der Broker fügt den Client zur Liste der Subscriber, wenn er sich für einen Topic interessiert und nimmt eine Nachricht entgegen, wenn er eine sendet, und reiht diese in die Liste der Nachrichten.

3.3.2 Distributor

Der Distributor prüft ständig die Liste von Messages nach Nachrichten, die zugestellt werden sollen. Jedes Mal, wenn eine Nachricht erfolgreich zugestellt wurde, wird den Zähler 'nattempts' um eins erhöht. Falls die Zustellung fehlschlägt, wird der Timestamp 'last fail' gesetzt. Mit diesem Timestamp kann der Distributor auch feststellen, ob er bereits einen erneuten Versuch zur unternehmen soll.

3.3.3 Garbage Collector

Der Garbage Collector räumt Statistiken, Nachrichten und Subscriber auf. Mittels bestimmter Regeln entscheidet der Garbage Collector, ob ein Eintrag in der Liste der Statistiken zu einer Nachricht gelöscht werden kann (z.B. Nachricht wurde erfolgreich zugestellt oder Subscriber hat die Verbindung beendet). Sobald eine Nachricht keine Statistiken mehr hat, kann die gesamte Nachricht abgeräumt werden und aus der Liste der Messages gelöscht werden. Wenn ein Subscriber in keiner der Liste der Statistiken mehr auftaucht, kann auch er gelöscht werden.

4 Implementation

5 Verwendung

6 Rückblick

7 Fazit

8 Anhang

8.1 Begriffe

Begriff	Erklärung
Topic	Nachrichten werden in Themen gruppiert. Clients abonnieren Themen und kriegen so die Nachrichten zugestellt, die an dieses Thema gesendet werden.
Publisher	Ein Client, der Nachrichten sendet
Subscriber	Ein Client, der an Nachrichten von einem Topic interessiert ist
STOMP	Simple Text Oriented Messaging Protocol