DHBW
Duale Hochschule
Baden-Württemberg
Heidenheim

Spätester Abgabetermin: 31.03.2023

Neue Datenbankkonzepte

Dokumentation zum Thema
„Redis / Lokalisierung für Rettungsdienste“

eingereicht

am **23.03.2023**

Modul:	**Datenbanken im Business Engineering Kontext**
Kurs:	WWI-2021-B
Semester:	WiSe 2022/23 (4)
Gruppenmitglieder:	Markus Hoffmann
	Michael Dolinac
	Johannes März
Dozent/Prüfer:	Marvin Scharle

Vorbemerkungen

Um einen besseren Lesefluss zu ermöglichen, wurde in dieser Arbeit auf die Verwendung gendergerechter Schreibweise verzichtet. Wird im Folgenden das generische Maskulinum verwendet, so schließt dies stets ausdrücklich auch weibliche sowie anderweitige Geschlechteridentitäten mit ein.

Inhaltsverzeichnis

Abbildungsverzeichnis

1. Projektvorstellung

Rettungskräfte auf der ganzen Welt haben alle mit dem gleichen Problem zu kämpfen: „Wie kommen wir am schnellsten zum Einsatzort?“. Ein integraler Bestandteil dieses Problems ist die Alarmierung der nächstgelegenen Einsatzkräfte. Im Rahmen dieses Projekts soll eine technische Lösung dieses Problems mit der NoSQL-Datenbank Redis realisiert werden. Ziel ist es, insbesondere die enorme Performance von Redis zu demonstrieren. Aber auch die für Rettungskräfte wichtige hohe Verfügbarkeit und Skalierbarkeit soll durch eine Redis-Cluster-Integration dargestellt werden.

2. Anforderungen

Die Rettungskräfte sollen selbstständig ihre Position und Kennung an eine REST-Schnittstelle übermitteln können. Diese Übertragung kann periodisch geschehen, beispielsweise alle drei Sekunden. Im Falle eines Notrufes wird die Position des Notrufers über „Automatic Mobile Location“ (AML) an die Notrufzentrale übermittelt. Mithilfe dieser Position kann dann an eine REST-Schnittstelle eine Abfrage gestellt werden, bei der in einem gewünschten Radius um diese Position alle Rettungskräfte ermittelt werden. Mittels dieser Information können schließlich die am nächsten gelegenen Rettungskräfte alarmiert werden.

3. Technische Umsetzung

Das Projekt wird mithilfe von NodeJS und Express realisiert. Für die geforderten Schnittstellen werden REST-Endpunkte mit der gewünschten Funktionalität erstellt. Redis ermöglicht es, mithilfe von vorgefertigten Geo-Spatial-Funktionen einen Geo-Spatial-Index anzulegen, unter dem die Positionen der einzelnen Rettungskräfte gespeichert werden können. Zu jedem Eintrag innerhalb des Indizes wird die Position in Form von Längen und Breitengrad gespeichert.

Um nun den tatsächlichen Einsatz der Schnittstellen und Redis zu simulieren, kann in einem separaten Worker-Thread ein Testdatengenerator gestartet werden, welcher eine in der „.env“-Datei bestimmte Anzahl von Rettungskräften dem Geo-Spatial-Index hinzufügt. Anschließend wird eine vom Nutzer bestimmte Anzahl von Simulatoren gestartet, welche asynchron und zufällig von einzelnen Rettungskräften die Positionen aktualisieren. Da dieser Vorgang in einem separaten Thread geschieht, wird der Main-Thread nicht blockiert und kann weiterhin die Anfragen über die REST-Schnittstellen entgegennehmen („non blocking architecture“).

4. Redis-Cluster

Um die bereits erwähnten Punkte der Hochverfügbarkeit und Skalierbarkeit zu realisieren, bietet die Node-Anwendung nicht nur Unterstützung einer eigenständigen Redis-Instanz an, sondern auch für einen Redis-Cluster. Hierbei werden die Daten mittels asynchroner Replikation über Master und Slaves repliziert. Dadurch ist sowohl eine hohe Verfügbarkeit als auch eine hohe Performance möglich. Bei einem Redis-Cluster müssen die Daten über die Knoten verteilt werden. Hierzu gibt es bei Redis-Cluster 16384 Hash-Slots, welche auf alle Master-Knoten im Cluster aufgeteilt werden. Um einen Key auf dem Cluster zu speichern, wird der Key (in diesem Fall der Geo-Spatial-Index) durch zyklische Redundanzprüfung (16-bit, XMODEM) gehashed und anschließend modulo (Teilen mit Rest) der Gesamtanzahl von Hash-Slots (16384) gerechnet.[1]

Das Ergebnis ist der Slot, in dem der Key mit seinen Werten gespeichert wird.

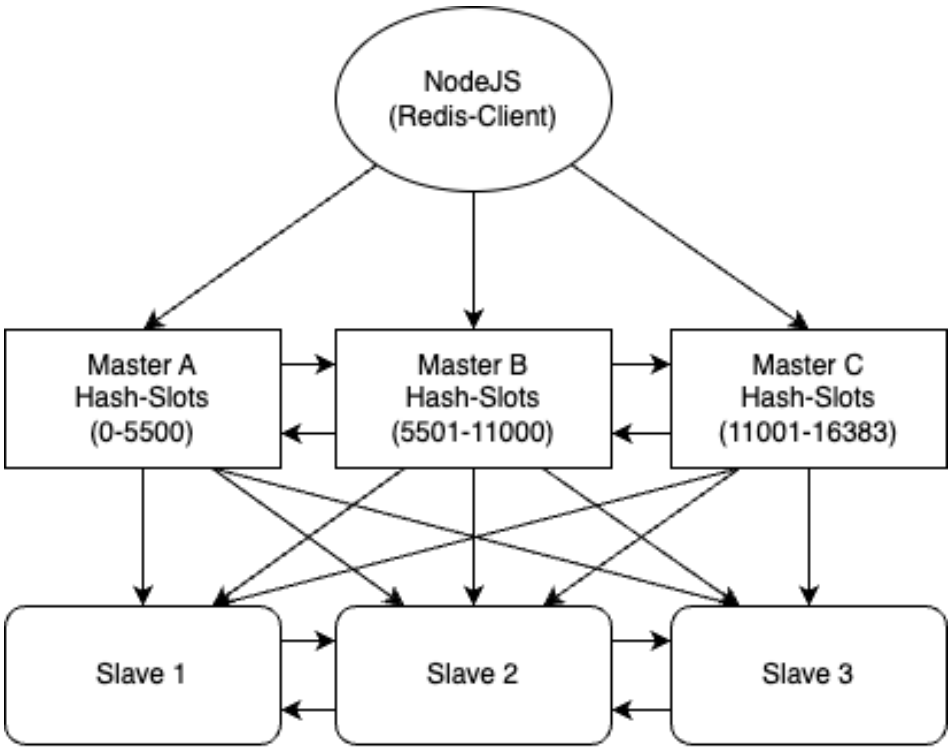

Abbildung 1: Beispiel Redis-Cluster mit Mindestkonfiguration von drei Masterknoten

[1] Vgl. Redis (2023); Vgl. Harston, J. G. (2013).

Beispiel:

CRC-16/XMODEM („rescueservice“) = 28089

28089 mod 16384 = 11705

Bei dem gewählten Beispiel-Cluster wird der Key „rescueservice“ dementsprechend auf dem Master C gespeichert, da dieser für die Hash-Slots 110001-16383 zuständig ist und 11705 in diesem Raum liegt.

Jedoch weist diese Implementierung ein Problem auf. Die Beispielanwendung speichert die gesamten Geo-Spatial-Informationen von jeder Rettungskraft in einem einzigen Index/Schlüssel. Dies ist prinzipiell so notwendig, um die Geo-Spatial-Funktionen auf die Gesamtmenge von Rettungskräften anwenden zu können. Dadurch werden allerdings sämtliche Schreiboperationen auf nur einem Master ausgeführt, wodurch die (Schreib-) Performance durch diesen limitiert wird. Um dieses Problem zu lösen, müsste man die Rettungskräfte auf mehrere Indizes/Schlüssel aufteilen, um so die Daten auf mehrere Master verteilen zu können. Hierzu eigenen sich Verfahren wie zum Beispiel Quadtrees oder spezielle Sifting-Methoden, welche die Geo-Spatial-Daten systematisch sharden, um sie auf dem Cluster verteilen zu können.[2]

5. Performance

Gemessen auf einem MacBook Pro 2021 16“ M1 Pro 16GB (10C/16GPU) erreicht der Simulator einen Durchsatz von ca. 250.000 Commands/s auf einer einzelnen Redis Instanz. Dies entspricht ca. 125.000 Positionsupdates pro Sekunde, da bei jedem Update eine möglichst „realistische“ neue Position ausgehend der ursprünglichen Position berechnet wird.

In einem Cluster ist die Performance aufgrund des Overheads der Replikation minimal schlechter (~2%), auch skaliert die Leistung aufgrund der beschriebenen Problematik nicht mit der Anzahl der Knoten im Cluster.

Um ein Gefühl für diese Zahlen zu bekommen, nachfolgend einige Daten, um sie einordnen zu können:

Anzahl von Krankenwagen und Notarzt-fahrzeugen in Deutschland (2022)[3]	**Anzahl von Feuerwehrfahrzeugen in Deutschland (2022)**[4]
22.300	2.873

[2] Vgl. Iyer, A. P. & Stoica, I. (2017); Vgl. Carmenta (2019).
[3] Vgl. Kords, M. (2022).
[4] Vgl. Deutscher Feuerwehrverband (2023).

(…) Wie die Daten zeigen, ist Redis trotz seiner Single-Threaded-Natur ideal dafür geeignet die Positionen aller Rettungskräften in Deutschland in nahezu Echtzeit zu speichern und zu verwerten, und das alles auf einer einzelnen Redis-Instanz. Wenn die beschriebene Problematik gelöst wird, können auf einem Redis-Cluster sogar nahezu unbegrenzt Geo-Spatial-Daten verarbeitet werden. Allerdings eignen sich dann die Redis eigenen Geo-Spatial-Funktionen nicht mehr, da diese nur auf einen Geo-Spatial-Index angewendet werden können.

6. Fazit

Redis ist perfekt dafür geeignet, um hoch performante Geo-Spatial-Operationen auf sich stetig ändernde Daten durchzuführen. Auch die vielen Random-Updates/Reads auf einer einzelnen Redis-Instanz sind extrem performant. Eine nahezu unbegrenzte horizontale Skalierung und Verfügbarkeit der Geo-Spatial-Daten sind dank einem Redis-Cluster möglich. Jedoch müssen ab gewissen Grenzen andere Sharding-Mechanismen eingesetzt werden, um Limitierungen einer einzelnen Instanz zu umgehen und alle Knoten eines Clusters gleichmäßig auslasten zu können.

Quellenverzeichnis

(nach Dokumentationsinhalt sortiert)

- Harston, J. G. (2013): *Calculating 16-bit CRCs (CRC-16)*.
  Online unter: https://mdfs.net/Info/Comp/Comms/CRC16.htm.
  Abfrage am 23.03.2023.

- Redis (2023): *Scaling with Redis Cluster*.
  Online unter: https://redis.io/docs/management/scaling/.
  Abfrage am 23.03.2023.

- Iyer, A. P. & Stoica, I. (2017): *A scalable distributed spatial index for the internet-of-things*.
  Online unter: https://dl.acm.org/doi/10.1145/3127479.3132254.
  Abfrage am 23.03.2023.

- Carmenta (2019): *Master's Thesis: Spatial indexing for moving geometry in main memory*.
  Online unter: https://carmenta.com/en/innovation-labs/spatial-indexing-for-moving-geometry-in-main-memory/.
  Abfmrage am 23.03.2023.

- Kords, M. (2022): *Anzahl der Krankenwagen und Notarzteinsatzfahrzeuge in Deutschland von 2012 bis 2022*.
  Online unter: https://de.statista.com/statistik/daten/studie/456644/umfrage/bestand-an-krankenwagen-in-deutschland/.
  Abfrage am 23.03.2023.

- Deutscher Feuerwehrverband (2023): *Statistik: Erfassung statistischer Daten*.
  Online unter: https://www.feuerwehrverband.de/presse/statistik/.
  Abfrage am 23.03.2023.

Anhang

- GitHub-Repository mit:
  - Quellcode
  - README.md (Anweisungen für Installation & Ausführung)