Hazelcast

by Alexander Edinger, Jonas Eppard and Fernand Hoffmann

TODO: General Introduction

1 Key-Value-Stores

TODO:

1.1 History of Key-Value-Stores

TODO: Change

1.2 Working principle of Key-Value-Stores

TODO:

1.3 Comparision of Key-Value-Stores to relational Databases

TODO:

2 Hazelcast in Theory

TODO:

2.1 Hazelcast in CAP-Theorem

TODO:

2.2 BASE in Hazelcast

TODO:

3 Hazelcast in Practice

The goal of this section is to test Hazelcast in practice. With this goal in mind, we will first install Hazelcast, and then we will try to implement some example data structures. We will take a look at the Usability of the Hazelcast interface.

In this section following limitations are taken into account: First only the Open-Source Hazelcast Platform is used. Second, only the Hazelcast Command Line Interface (CLI) is used. Therefore, the Hazelcast cloud service Viridian is not used and will not be evaluated. Furthermore, the Hazelcast Clients outside the Hazelcast CLI are not used, and will not be evaluated. The data structure used for testing is taken from a previous lecture. TODO:

3.1 Installation of Hazelcast

There are multiple ways to install Hazelcast. Due to the fact that Hazelcast is open-source and provides Binaries. Therefore, it's possible to build the Hazelcast Platform from source. However, it is easier to download prebuild binaries. For Linux there is a Debian package available. For macOS there is a brew package available, but for Windows there is no package or installer available. Therefore, under Windows you have to download prebuild binaries, if you don't want to build Hazelcast from source. Another option is to use docker. (Hazelcast, 2023b)

We tried the Hazelcast installation on multiple operating systems, but weren't able to test the installation on macOS. Therefore, we will only describe the installation on Linux and Windows, and will take a look at the installation via docker. The installation on Linux is very easy and straight forward: First you will download the public key of the Hazelcast repository and add the repository to your package manager sources:

```
wget -qO - https://repository.hazelcast.com/api/gpg/key/public | gpg
   --dearmor | sudo tee /usr/share/keyrings/hazelcast-archive-keyring
   .gpg > /dev/null
echo "deb [signed-by=/usr/share/keyrings/hazelcast-archive-keyring.
   gpg] https://repository.hazelcast.com/debian stable main" | sudo
   tee -a /etc/apt/sources.list
```

Listing 1: Adding the Hazelcast repository to the package manager sources under Linux (Debian) (Hazelcast, 2023b)

After that you will be able to install Hazelcast via the package manager like any other package:

```
sudo apt update && sudo apt install hazelcast
```

Listing 2: Installing Hazelcast under Linux (Debian) (Hazelcast, 2023b)

The installation under Windows was a bit more complicated. First Hazelcast did not provide a Windows installer. Therefore, we had to download the binary from the Hazelcast website. This binary wasn't able to run directly, because it needed the `JAVA_HOME` environment variable to be set. First we tried to install a Java Runtime Environment (JRE) and set the `JAVA_HOME` variable accordingly. However, the JRE was not able to run the Hazelcast binary and a Java Development Kit (JDK) was needed. Therefore, we installed the JDK set the path, and then the Hazelcast binary was able to run. However, personally we found the installation with docker the easiest. First, the Hazelcast image is downloaded from the docker hub:

```
docker pull hazelcast/hazelcast
```

Listing 3: Downloading the Hazelcast image from the docker hub (Hazelcast, 2023b)

After that the local Hazelcast Cluster is started as specified in the documentation:

```
docker network create hazelcast-network
docker run \
    -it \
    --network hazelcast-network \
```

```
--rm \
-e HZ_CLUSTERNAME=hello-world \
-p 5701:5701 hazelcast/hazelcast
```

Listing 4: Starting the Hazelcast Cluster using the docker image (Hazelcast, 2023c)

For the following example the docker installation will be used.

3.2 Example in Hazelcast

In this section an example data structure will be implemented using the Hazelcast CLI. The Hazelcast CLI is run as a docker container. The data structure is taken from a previous lecture and shown in Fig. 1. The data

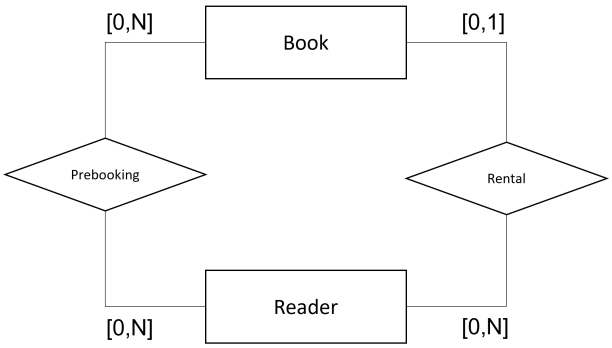

Figure 1: Example data structure (see also Mairhofer, 2021)

structure models a library with books. The library has multiple books and multiple readers. Each book can be rented by one reader. The reader can rent multiple books. And if a book is rented by a reader, another reader can't rent it, but order a prebooking for it. This data structure was given to us in an example in a previous lecture by Mairhofer (2021). The exact fields of the data structure are not important for this example. The important part is that the data structure requires foreign keys to model the relations.

Hazelcast is a distributed key-value store, as described in Section 2. It does not provide a relational database, and it has no support for foreign keys. It is possible to implement relations with embedded objects or with user defined fields. To define the Relations between the object classes we will use user defined fields. Embedded objects have the problem, that there are hard to manage by hand and difficult to implement many-to-many relations. In the user defined fields we can define the fields like a foreign

key in a relational database, but Hazelcast does not provide any safety checks. Therefore, a user has to implement safety checks by hand, if the consistency in the relations are important. In this example we will not write an application and only take a look at the Hazelcast CLI. Therefore, we will not implement any safety checks. But Hazelcast provides a way to select and join different maps. Thus, it is possible to implement the relations and query those relations inside the Hazelcast CLI.

To conclude our example we have successfully implemented a data structure with foreign keys in the key-value store Hazelcast. However, the relations are not checked by the Hazelcast platform and the user has to implement the checks to ensure that the relations are consistent. Therefore, we advise using Hazelcast as a key-value store and not to try using the Hazelcast platform to implement complex relations.

4 Evaluation of Hazelcast

TODO:

4.1 Advantages of Hazelcast

TODO:

4.2 Disadvantages of Hazelcast

TODO:

4.3 Lessons learned

This paragraph contains a short summary of the lessons learned during the evaluation of Hazelcast in Practice. First, Hazelcast is easiest to install via docker. Secondly the Hazelcast CLI is easy to use with common SQL like syntax, but has some limitations and differences. Mainly in Hazelcast the

common concept is a map (Mapping) instead of a table. As a user you have to create mappings and not tables. After that you should take a look at the Hazelcast Client APIs to use Hazelcast from within your application. These Clients are currently available for Java, .NET, C++, Node.js, Python and Go (Hazelcast, 2023a).

5 Conclusion and Recommendations

TODO: Conclusion and Recommendations

References

Ab architect's view of the hazelcast platform [Medium: White Paper]. (2023). Retrieved March 9, 2023, from https://hazelcast.com/thank-you/?resource_id=2006&form_page=https%3A%2F%2Fhazelcast.com%2Fresources%2Farchitects-view-hazelcast%2F

Brewer, D. E. A. (2000). Towards robust distributed systems.

Coda Hale. (2010, October 7). *You can't sacrifice partition tolerance* [Codahale.com]. Retrieved March 13, 2023, from https://codahale.com//you-cant-sacrifice-partition-tolerance/

Haerder, T., & Reuter, A. (1983). Principles of transaction-oriented database recovery. *ACM Computing Surveys*, *15*(4), 287–317. https://doi.org/10.1145/289.291

Hazelcast. (2022a). *Hazelcast*. Retrieved March 7, 2023, from https://github.com/hazelcast

Hazelcast. (2022b). *Hazelcast/hazelcast* [GitHub]. Retrieved March 7, 2023, from https://github.com/hazelcast/hazelcast

Hazelcast. (2022c). *SQL*. Retrieved March 9, 2023, from https://docs.hazelcast.com/hazelcast/latest/sql/sql-overview

Hazelcast. (2022d, November 3). *What is the hazelcast platform? | hazelcast explainer*. Retrieved March 9, 2023, from https://www.youtube.com/watch?v=UpIgHzKbMp0

Hazelcast. (2023a). *Clients & languages* [Hazelcast]. Retrieved March 13, 2023, from https://hazelcast.com/clients/

Hazelcast. (2023b). *Installing hazelcast open source*. Retrieved March 7, 2023, from https://docs.hazelcast.com/hazelcast/latest/getting-started/install-hazelcast

Hazelcast. (2023c). *Start a local cluster in docker*. Retrieved April 8, 2023, from https://docs.hazelcast.com/hazelcast/latest/getting-started/get-started-docker

Mairhofer, K. (2021, April 14). Aufgabe SQL Buchausleihe.