



ONDERZOEKSVERSLAG

NOSQL database



JANUARY 16, 2020

HAN

Ivan Miladinovic 599294

Contents

1. Inleiding.....	2
2. Database	3
2.1 Redis.....	3
2.2 MongoDB.....	3
3. Implementatie	4
4. Conclusie	8
Literatuurlijst.....	9

1. Inleiding

In dit onderzoeksverslag wordt er onderzoek gedaan naar een NOSQL database voor het ophalen van gegevens. Deze gegevens worden gebruikt om rapportages te maken.

De opdracht binnen dit project betreft het ophalen van rapportages. Dit moet “snel” gaan. Het ophalen van een rapportage mag niet langer dan een seconde duren.

In combinatie met onze doelstelling is daarom de volgende hoofdvraag opgesteld: “Welke NoSQL database is het meest geschikt voor deze opdracht?”

Met de volgende deelvragen:

- Wat is MongoDB?
- Wat is Redis?
- Welke database haalt gemiddeld het snelste de gegevens op?

2. Database

In dit hoofdstuk wordt door de projectgroep uitgekozen databases nader bekeken. De databases zijn gekozen gebaseerd op het advies van de procesbegeleider. De kandidaten zijn als volgt: Redis en MongoDB.

2.1 Redis

Redis is een opensource in-memory key-value database. In-memory houdt in dat de gegevens in het geheugen van de computer opgeslagen wordt. In-memory databases zijn sneller dan SQL databases die de gegevens in een harddisk of solid state drive doen omdat het de gegevens opslaat in het geheugen. (Redis, 2020)

Bij Redis worden de gegevens opgeslagen als key-value of te wel als sleutels met waarden. Hiermee worden de gegevens opgehaald met de sleutel en zit de informatie in de waarde van de sleutel.

Redis heeft als voordeel dat het ophalen van gegevens razendsnel is omdat de gegevens zich in het geheugen bevindt.

Nadeel hiervan is dat de gegevens niet opgeslagen wordt waardoor een verlies van stroom de gegevens in Redis verloren gaan.

2.2 MongoDB

MongoDB is een opensource document-georiënteerde database gebouwd voor moderne applicatieontwikkelaars. Document houdt in dat de gegevens in de vorm van BSON (Binear JSON) worden opgeslagen waardoor het een flexibele structuur heeft. Waar in SQL de gegevens in een tabel worden opgeslagen, heet dit bij MongoDB een "collection". (MongoDB, 2020) MongoDB slaat standaard de informatie op in een hardeschijf of solid state drive.

MongoDB bevat een aantal voordelen. Het is zeer flexibel. Gegevens zijn niet gedefinieerd dus kan je meerdere datatypen in een collection opslaan. Informatie uit MongoDB hoeft ook niet geconverteerd te worden naar een object in de applicatie. MongoDB maakt gebruik van Storage Engines. Dit is een component van de database die verantwoordelijk is voor het beheren hoe data opgeslagen wordt. Dit betekent dat MongoDB met een andere Storage Engine de gegevens volledig in het geheugen kan opslaan. Hierdoor heeft het dezelfde voordelen en nadelen als Redis. (Storage Engines, 2020)

MongoDB is populair waardoor er veel documentatie over te vinden is.

Het nadeel van MongoDB is dat gegevens meer opslag innemen omdat er geen gebruik wordt gemaakt van normalisatie.

3. Implementatie

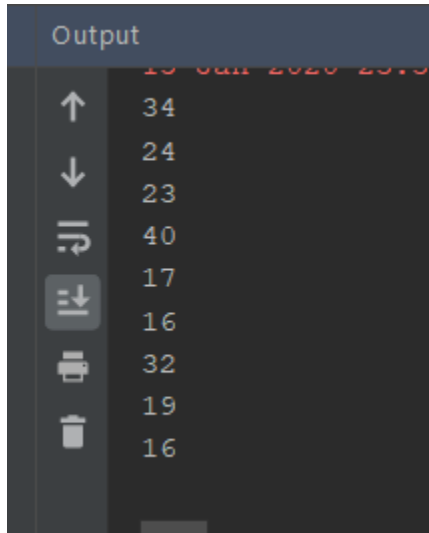
Hier implementeren we de 3 databases zodat we de prestaties kunnen vergelijken. Hierbij testen we de SQL, MongoDB en Redis implementaties. We gebruiken de API om te testen hoe lang het duurt voordat een rapportage is opgehaald. We vergelijken de resultaten hiervan om tot een conclusie te komen.

Bij het testen gebruiken we de volgende code:

```
public Rapportage getRapportage(String rapportageNaam) {  
    Rapportage rapportage = new Rapportage();  
    long startTime = System.currentTimeMillis();  
    rapportage.setRapportage(rapportageDAO.getRapportageSQL(rapportageNaam));  
    long endTime = System.currentTimeMillis();  
    System.out.println((endTime - startTime));  
    return rapportage;  
}
```

Figuur 1

Met deze test wordt er gekeken hoeveel tijd er verstreken is met het ophalen van de rapportage. Eerst krijgt de variabel `startTime` de huidige tijd van het systeem in milliseconden. Daarna wordt de functie uitgevoerd om het uit de SQL database te halen. Vervolgens krijgt de variabel `endTime` de huidige tijd van het systeem. Het resultaat wordt in het console venster weergegeven, zie figuur 2. Het resultaat is het aantal milliseconden die verstreken zijn.



Figuur 2 console venster

Voor het ophalen van een rapportage zijn er drie functies gemaakt:

```
public String getRapportageSQL(String rapportage){
    String rapportageJSON = "";
    try (Connection connection = connectionFactory.getConnection();
        PreparedStatement statement = connection.prepareStatement( sql: "SELECT (SELECT * FROM "+rapportage+" FOR JSON AUTO) AS Rapportage")) {
        ResultSet resultSet = statement.executeQuery();
        if(resultSet.next()){
            rapportageJSON = resultSet.getString( columnLabel: "Rapportage");
        }
        return rapportageJSON;
    } catch (SQLException e) {
        logger.log(Level.SEVERE, msg: "SQL EXCEPTION BIJ OPHALEN VAN RAPPORTAGES", e);
        throw new RuntimeException(e);
    }
}
```

Figuur 3 SQL rapportage

```
public String getRapportage(String rapportage){
    String rapportageJson = jedis.get(rapportage);
    jedis.close();
    return rapportageJson;
}
```

Figuur 4 Redis rapportage

```
public String getRapportageMongo(String rapportage){
    String rapportageJson = "";
    MongoClient<Document> collection = connectionFactory.getDatabase().getCollection( s: "rapportage");
    BasicDBObject query = new BasicDBObject();
    query.put("rapportageNaam", rapportage);
    MongoClient<Document> cursor = collection.find(query).iterator();
    try {
        while (cursor.hasNext()) {
            Document result = cursor.next();
            rapportageJson = (String)result.get("rapportage");
        }
        return rapportageJson;
    } finally {
        cursor.close();
    }
}
```

Figuur 5 MongoDB rapportage

Bij SQL moet er een specifiek query worden geschreven waarbij de rapportage opgehaald wordt met een view.

Het ophalen gaat met Redis het makkelijkste aangezien Redis alleen de sleutel hoeft op te zoeken.

Bij MongoDB moeten we eerst in een specifiek collection zoeken. Daarna wordt de naam van de rapportage gezocht. Het resultaat van de waarde wordt omgezet na een JSON string.

Om te testen halen we de rapportage "RAPPORTAGE_omzet_week_filiaal" van week 10, 11 en 12 ieders driemaal op.

De gegevens van de rapportage zien er als volgt uit:

	Vestigingsnaam	Woonplaats	aantal_cheques	omzet
1	Dopbanilazz International Group	Gullane	1	23.00
2	Empebedover Company	Isle of Bute	3	144.00
3	Froerplantor Holdings Company	Leigh-on-Sea	3	87.00
4	Inpebicover Inc	Yelverton	1	36.00

Figuur 6 test rapportage

Bij het testen kwamen de volgende resultaten:

SQL:

Rapportage	Tijd in ms 1e	Tijd in ms 2e	Tijd in ms 3e	Gemiddeld in ms
Week 10	34	24	23	27
Week 11	40	17	16	24,3
Week 12	32	19	16	22,3

Figuur 7 SQL resultaten

Redis:

Rapportage	Tijd in ms 1e	Tijd in ms 2e	Tijd in ms 3e	Gemiddeld in ms
Week 10	27	4	2	11
Week 11	2	3	3	2,66
Week 12	2	3	3	2,66

Figuur 8 Redis resultaten

MongoDB:

Rapportage	Tijd in ms 1e	Tijd in ms 2e	Tijd in ms 3e	Gemiddeld in ms
Week 10	119	11	9	46,3
Week 11	10	8	8	8,66
Week 12	8	11	9	9,33

Figuur 9 MongoDB resultaten

Voor het testen wordt de API opnieuw opgestart. Dit om de test zo eerlijk mogelijk te maken. Dit is terug te zien bij het eerste resultaat van week 10 waarbij elke database trager is dan gemiddeld. Hierin is MongoDB het traagst maar dit kan te maken hebben met de implementatie.

Hier is te zien dat SQL gebruik maakt van Indexes. Bij het begin van elk rapportage is SQL slomer dan gemiddeld. Dit heeft te maken met het execution plan. Bij het eerste keer ophalen van de query wordt de beste manier bepaald om de query uit te voeren. Na het ophalen slaat SQL de execution plan op in het geheugen. Hierdoor zijn de volgende resultaten sneller opgehaald. (Little, 2016)

Uit de resultaten blijkt dat Redis de snelste uit de test komt. Vergeleken met SQL biedt Redis een prestatieverbetering van gemiddeld een factor 5. Zie figuur 8. Bij figuur 9 is te zien dat MongoDB met 8 resultaten gemiddeld een factor of twee sneller is dan SQL. MongoDB staat hierdoor op tweede plek.

4. Conclusie

Op basis van dit onderzoek is gekozen voor het gebruik van Redis, met als voornaamste reden de prestaties. Doordat de gegevens worden opgeslagen in het geheugen, is het ophalen van gegevens razendsnel. Hierdoor worden de rapportages binnen een seconde opgehaald. Dit zie je terug in de resultaten en conclusie van de test in het vorige hoofdstuk. Met deze onderzoek is mijn hoofdvraag beantwoordt.

Literatuurlijst

MongoDB. (2020, Januari 15). *What is MongoDB?* Opgeroepen op Januari 15, 2020, van MongoDB: <https://www.mongodb.com/what-is-mongodb>

Redis. (2020, Januari 4). *Redis Homepage*. Opgeroepen op Januari 4, 2020, van Redis: <https://redis.io/>