Concurrency in NoSQL Databases

Multidisciplinair Project 19/20: Thomas Muller

# Inleiding

Omdat we tijdens ons project hoogstwaarschijnlijk gebruik gaan maken van een database, is het geen overbodige luxe om te weten hoe we om moeten gaan met concurrency. Tijdens de hoofdfase van de opleiding HBO-ICT aan de HAN wordt concurrency in een relationele database al behandeld. Hierom wordt in dit onderzoek gekeken naar hoe dit kan worden gedaan in een NoSQL database.

## Hoofdvraag:

* Op welke manier beïnvloedt concurrency een NoSQL database en hoe kan hier het beste mee worden omgegaan?
* Wat is een voorbeeld van de implementatie van de ‘best practices’ voor het omgaan met concurrency in een NoSQL database?

## Deelvragen

* Hoe verschilt concurrency in een NoSQL database van concurrency in een relationele database?
* Welke gereedschappen bieden NoSQL databases om concurrency control uit te voeren?
* Wat is een mogelijke implementatie concurrency control binnen NoSQL waar twee gebruikers een gedeelde resource tegelijkertijd willen aanpassen?

# Deelvraag 1: Hoe verschilt concurrency in een NoSQL database van concurrency in een relationele database?

concurrency is de mogelijkheid van een database om meerdere gebruikers meerdere transacties tegelijkertijd te laten beïnvloeden. Dit is een van de belangrijkste eigenschappen die een database scheidt van andere vormen van dataopslag, zoals spreadsheets. De mogelijkheid om concurrency aan te bieden is uniek voor databases.

Dit kan echter problemen met zich meebrengen als er niet goed is nagedacht over hier mee wordt omgegaan.

Voor dit onderzoek zal ik gebruik maken van MongoDB[[1]](#footnote-0).

MongoDB maakt gebruik van een variant van optimistic control, genaamd multi-granularity locking[[2]](#footnote-1) om consistentie binnen de database te waarborgen. Dit houdt in dat er naast shared (S) en exclusive (X) locks ook intent shared (IS) en intent exclusive (IX) locks zijn. Als een row of collection wordt gelocked worden alle hogere niveaus vergrendeld met een intentlock.

Bijvoorbeeld, wanneer een collection wordt vergrendeld om met een write lock (X), moeten zowel het hogere bijbehorende databaselock als het globale slot worden vergrendeld in de exclusive intent (IX) -modus. Een database kan tegelijkertijd worden vergrendeld in de IS- en IX-modus, maar een exclusief (X) slot kan niet samengaan met andere locks. Een shared (S) slot kan alleen naast elkaar bestaan met intent shared (IS) sloten.

Deze hiërarchische manier van locking past goed bij een database zoals MongoDB door de opzet van databases, collections en documents.

MySQL zou een First-In-First-Out (FIFO) aanpak gebruiken, waar de locks een voor een worden toegewezen op basis van volgorde.

Mongo doet dit anders; Alle locks die compatibel zijn worden gegrant en gereleased waar mogelijk. Een S lock zou dus voor een X lock gegrant kunnen worden ook al was het X lock eerder gequeue’d. Dit is betekent dus dat dat S lock compatibel was met de huidige lock en dat X lock niet.

# Deelvraag 2: Welke gereedschappen bieden NoSQL databases om concurrency control uit te voeren?

# Deelvraag 3: Wat is een mogelijke implementatie concurrency control binnen NoSQL waar twee gebruikers een gedeelde resource tegelijkertijd willen aanpassen?

# 

1. https://www.mongodb.com [↑](#footnote-ref-0)
2. https://docs.mongodb.com/manual/faq/concurrency/#mgl-ref [↑](#footnote-ref-1)