11-6-2023

Yanniek Wielage

NHL Stenden

Netflix Database

Onderzoeksdocument

# Inleiding

Netflix is in 1997 opgericht door Reed Hastings en Marc Randolph met als primaire doel het verhuren van DVD’s in Amerika. In 2007 begon Netflix met het streamen van media en in 2010 heeft Netflix ervoor gekozen om uit te breiden naar Canada. Er is de laatste jaren hard gewerkt om de streaming-dienst ook uit te rollen naar de rest van de wereld. Netflix gebruikt op dit moment een Oracle database die opgezet is op het moment dat de streaming-dienst (destijds nog een dvd-rental-service) gelanceerd werd. De organisatie heeft twijfels over de extra capaciteit die nodig is om het systeem soepel te houden na internationalisering. Daarom overweegt Netflix om opnieuw kritisch te kijken naar het huidige database ontwerp en de geïmplementeerde technieken. Netflix heeft Stenden benaderd om te onderzoeken of hedendaagse technieken een positieve bijdrage kunnen leveren aan de huidige implementatie.

Er zijn voor de gevraagde technieken testen uitgevoerd. De geteste technieken zijn:

* ADO.NET
* Entity Framework
* MongoDB

Alle technieken zijn uitgevoerd in C#, gebruik makend van Visual Studio. De snelheden van de CRUD-operaties bij alle drie technieken getest. Binnen dit onderzoeksdocument zijn de resultaten te vinden.

Omdat 1,000,000 rijen te lang zal duren, is er op dit punt terug geschaald. Er is gekozen voor 10, 50 en 100 rijen voor de testen. Dit aantal is gekozen omdat er een limitatie is bij het testen van MongoDB. Bij het testen van 1000 wou MongoDb niet meer meewerken, waarschijnlijk door een limitatie aan aanvragen bij het gratis abonnement. Dit abonnement is gebruikt bij dit onderzoek. Voor elke rijaantal zijn er drie testen gedaan, voor vertrouwbaarheid van de data.

Bij dit onderzoek is er gebruik gemaakt van de volgende specificaties:

**Moederbord: MSI Tomahawk MAX**

**Grafische kaart NVIDIA Geforce RTX 2070 Super**

**RAM: 16,0 GB**

**Processor: AMD Ryzen 7 3700X 8-Core Processor 3.60 GHz**

# MongoDB

## CREATE

|  |  |  |  |
| --- | --- | --- | --- |
| Test/Aantal | 1 | 2 | 3 |
| 10 | 00:00:00.0821079 | 00:00:00.0843676 | 00:00:00.0686908 |
| 50 | 00:00:00.0784294 | 00:00:00.0843722 | 00:00:00.0807040 |
| 100 | 00:00:00.0933295 | 00:00:00.0973571 | 00:00:00.0946937 |

## READ

|  |  |  |  |
| --- | --- | --- | --- |
| Test/Aantal | 1 | 2 | 3 |
| 10 | 00:00:00.0357245 | 00:00:00.0356555 | 00:00:00.0345132 |
| 50 | 00:00:00.0404584 | 00:00:00.0430043 | 00:00:00.0406973 |
| 100 | 00:00:00.1467080 | 00:00:00.1531491 | 00:00:00.0821048 |

## UPDATE

|  |  |  |  |
| --- | --- | --- | --- |
| Test/Aantal | 1 | 2 | 3 |
| 10 | 00:00:00.1953184 | 00:00:00.3273287 | 00:00:00.2391779 |
| 50 | 00:00:03.7395206 | 00:00:01.0013615 | 00:00:01.0708139 |
| 100 | 00:00:01.8820858 | 00:00:02.2909851 | 00:00:02.0263694 |

## DELETE

|  |  |  |  |
| --- | --- | --- | --- |
| Test/Aantal | 1 | 2 | 3 |
| 10 | 00:00:00.2133705 | 00:00:00.3083303 | 00:00:00.2194364 |
| 50 | 00:00:01.0154062 | 00:00:00.9748325 | 00:00:00.8896295 |
| 100 | 00:00:02.0777239 | 00:00:02.0798396 | 00:00:01.9643804 |

**Toelichting**

Op het eerste gezicht valt op dat MongoDB bij kleine aantallen erg snel is. Wel valt het op dat hoe hoger het aantal documenten, hoe langer de operatie duurt. Dit valt veel meer op bij MongoDB. Ook was bij een update een geval die veel langer duurde dan de rest van de testen. Dit geval is gemarkeerd in het rood. Dit kan komen doordat er altijd een internetconnectie moet zijn tussen MongoDB en de ontvanger van de data.

# ADO.NET

## CREATE

|  |  |  |  |
| --- | --- | --- | --- |
| Test/Aantal | 1 | 2 | 3 |
| 10 | 00:00:00.1532579 | 00:00:00.1491673 | 00:00:00.1611160 |
| 50 | 00:00:00.1598219 | 00:00:00.1557296 | 00:00:00.1556932 |
| 100 | 00:00:00.1736152 | 00:00:00.1683661 | 00:00:00.1831918 |

## READ

|  |  |  |  |
| --- | --- | --- | --- |
| Test/Aantal | 1 | 2 | 3 |
| 10 | 00:00:00.1865082 | 00:00:00.1860047 | 00:00:00.1854580 |
| 50 | 00:00:00.2240765 | 00:00:00.2214981 | 00:00:00.2401405 |
| 1000 | 00:00:00.2750389 | 00:00:00.3076407 | 00:00:00.2893808 |

## UPDATE

|  |  |  |  |
| --- | --- | --- | --- |
| Test/Aantal | 1 | 2 | 3 |
| 10 | 00:00:00.1494281 | 00:00:00.1567609 | 00:00:00.1520650 |
| 50 | 00:00:00.1555057 | 00:00:00.1560702 | 00:00:00.1645024 |
| 100 | 00:00:00.1648422 | 00:00:00.1722993 | 00:00:00.1580622 |

## DELETE

|  |  |  |  |
| --- | --- | --- | --- |
| Test/Aantal | 1 | 2 | 3 |
| 10 | 00:00:00.1603612 | 00:00:00.1553058 | 00:00:00.1663501 |
| 50 | 00:00:00.1727384 | 00:00:00.1948902 | 00:00:00.1619178 |
| 100 | 00:00:00.2412814 | 00:00:00.1915153 | 00:00:00.1890253 |

**Toelichting**

ADO.NET is als snelste uit de verf gekomen. Dit heeft waarschijnlijk te maken met het geval dat de database lokaal is. Ook valt het op dat er geen grote verschillen tussen de testresultaten zitten. ADO.NET lijkt erg consistent te zijn.

# Entity Framework

## CREATE

|  |  |  |  |
| --- | --- | --- | --- |
| Test/Aantal | 1 | 2 | 3 |
| 10 | 00:00:01.6872423 | 00:00:01.7323616 | 00:00:01.8274678 |
| 50 | 00:00:01.7523065 | 00:00:01.6957776 | 00:00:01.7180620 |
| 100 | 00:00:01.7593195 | 00:00:01.8105048 | 00:00:01.7348111 |

## READ

|  |  |  |  |
| --- | --- | --- | --- |
| Test/Aantal | 1 | 2 | 3 |
| 10 | 00:00:01.5862121 | 00:00:01.6328439 | 00:00:01.6569097 |
| 50 | 00:00:01.7036546 | 00:00:01.6811985 | 00:00:01.6652592 |
| 100 | 00:00:01.6683607 | 00:00:01.7299188 | 00:00:01.6929998 |

## UPDATE

|  |  |  |  |
| --- | --- | --- | --- |
| Test/Aantal | 1 | 2 | 3 |
| 10 | 00:00:01.6884411 | 00:00:01.7009786 | 00:00:01.7562770 |
| 50 | 00:00:01.7332538 | 00:00:01.6780212 | 00:00:01.7359174 |
| 100 | 00:00:01.6922776 | 00:00:01.7386651 | 00:00:01.6859816 |

## DELETE

|  |  |  |  |
| --- | --- | --- | --- |
| Test/Aantal | 1 | 2 | 3 |
| 10 | 00:00:01.6723467 | 00:00:01.6914963 | 00:00:01.7527274 |
| 50 | 00:00:01.6931172 | 00:00:01.7259051 | 00:00:01.7958506 |
| 100 | 00:00:01.7357476 | 00:00:01.7225785 | 00:00:01.7615357 |

**Toelichting**

Entity Framework (EF) is compleet code-based, wat het grote verschil is tussen ADO.NET en EF. Het valt vooral op hoeveel langzamer de CRUD-operaties zijn in vergelijking met ADO.NET, aangezien beide databases lokaal zijn.

# Conclusie

Er kan geconcludeerd worden dat ADO.NET de beste optie is voor een database op het gebied van snelheid. Wel kan er veel gezegd worden over MongoDB, aangezien NoSQL de data op een andere manier opslaat. Dit kan als een voordeel worden gezien. Ook is MongoDB volledig online en kan dat ook een belangrijk argument zijn om de keuze voor deze database te maken. Entity Framework komt als minste uit de test. Wel is Entity Framework het meest overzichtelijk voor een persoon die verstand heeft van code.