



Onderzoek Innovatie binnen Schaalbaarheid

Status: Definitief

Versie: 2.0

Juni 2021

VERSIEBEHEER

VERSIES

VERSIE	DATUM	AUTEUR	WIJZIGINGEN	STATUS
1.0	08-02-2021	Jay van Helderer	Opzet template	WIP
1.1	20-05-2021	Sander Berntsen	H1 t/m H4	WIP
1.2	03-06-2021	Sander Berntsen, Luuk Vermeer, Simon Westerburger	Inleiding, Werkplaats, Veldwerk, Showroom, Advies	Concept
2.0	04-06-2021	Sander Berntsen, Luuk Vermeer, Simon Westerburger	Feedback verwerken	Definitief

VERSPREIDING

VERSIE	DATUM	INGELEVERD BIJ
1.2	03-06-2021	Fred Veldmeijer
2.0	04-06-2021	Canvas

INHOUDSOPGAVE

1. Inleiding	4
2. Onderzoeksstrategieën.....	5
3. Bieb.....	6
Gevonden Schaalbaarheid Tools	6
Kubernetes	6
Qlik Scalability Tools	6
Loadview.....	6
NoSQL.....	6
Redis.....	6
Solr	6
Apache Zookeeper.....	7
Elastic Search	7
CloudFlare.....	7
Varnish	7
Amazon Load balancer	7
Gekozen Schaarbaarheid Tools.....	8
Elastic Search	8
Varnish	8
Redis met NoSQL.....	8
Loadview.....	8
4. Werkplaats	9
Elastic Search	9
Varnish	9
Loadview	10
Redis met NoSQL	13
5. Veldwerk	14
Varnish	14
Conclusie	14
LoadView	15
Conclusie	18
Redis met NoSQL	19
Conclusie	21
6. Showroom.....	22

Advies	23
Literatuurlijst	24

1. Inleiding

Wij werken met onze subgroep aan 2 onderzoeksvragen. Met het DOT framework proberen we deze vragen te beantwoorden. In dit document wordt de vraag beantwoord: Welke tools kunnen innovatie brengen op het gebied van schaalbaarheid? Dit document proberen we door het semester heen iteratief aan te vullen. Aan het eind van het semester hebben we een conclusie op de vraag "Waar binnen het agenda systeem kunnen tools, innovatie brengen op het gebied van schaalbaarheid?"

2. Onderzoeksstrategieën

Waar binnen het agenda systeem kunnen tools, innovatie brengen op het gebied van schaalbaarheid?

Bieb - Onderzoeken welke tools beschikbaar zijn om op schaalbaarheid te innoveren.

Werkplaats - POC's (Proof of Concept) maken van de tool(s)

Veldwerk - Achterhalen welke tools het beste werken d.m.v. A/B Testing.

Showroom - Tool(s) demonstreren aan de klant/docent d.m.v. een pitch en vragen voor feedback.

Stappenplan:

Bieb onderzoek voor het achterhalen welke tools er zijn.

Iedereen kiest een tool om uit te werken (3 in totaal).

Iedereen maakt een POC van zijn eigen tool.

De gemaakte POC met elkaar vergelijken d.m.v. A/B testen.

In showroom de tool demonstreren die uiteindelijk het beste eruit kwam.

3. Bieb

Gevonden Schaalbaarheid Tools

Hieronder zijn de bekendste tools weergegeven die wij konden vinden, die gebruikt kunnen worden voor het schaalbaar maken van software:

Kubernetes

Kubernetes is een portable, scalable, open-source platform voor het beheren van gecontaineriseerde workloads en services, dat zowel declaratieve configuratie als automatisering mogelijk maakt. (Kubernetes, sd)

Qlik Scalability Tools

Qlik Scalability Tools is een bundel van verschillende tools die gebruikt kan worden voor het testen van de performance en de schaalbaarheid van software. Een paar van de use cases hiervan zijn availability testing, stress testing, pre-caching, regression testing en hardware benchmarking. (Qlik)

Loadview

Loadview is een software dat gebruikt kan worden om websites te testen door het simuleren van website bezoekers. Loadview is niet open-source en heeft een downloadbare gratis demo. Enkele voorbeeld features van loadview zijn: gebruikers gedrag aanpassen, locatie simuleren, monitoren van de laadtijden, etc. (LoadView, sd)

NoSQL

NoSQL-databases doorbreken de traditionele mentaliteit van het opslaan van gegevens op één locatie. In plaats daarvan distribueert en slaat NoSQL gegevens op over een set van meerdere servers. Deze distributie van gegevens helpt de NoSQL-database server om de belasting te verdelen. (MongoDB, sd)

Redis

Redis is een Open Source in-memory data structure store die gebruikt wordt als database, cache en message-broker. Ook zorgt het voor automatische opdeling m.b.v. een Redis Cluster. (Redis, sd)

Solr

Solr heeft vanuit zichzelf geen mogelijkheden voor het schaalbaar opstellen van Software. Hiervoor wordt gebruik gemaakt van Apache Zookeeper. (Apache Solr, sd)

Apache Zookeeper

ZooKeeper is een open source Apache-project dat een gecentraliseerde service biedt voor het leveren van configuratie-informatie, naamgeving, synchronisatie en groepsservices over grote clusters in gedistribueerde systemen. Zookeeper wordt ook gebruikt in het Message Queuing software genaamd Kafka. (ZooKeeper, sd)

Elastic Search

Met Elasticsearch (ELK) kun je grote hoeveelheden gegevens snel en in bijna real-time opslaan, doorzoeken en analyseren in milliseconden. Het is in staat om snelle zoekreacties te krijgen omdat het in plaats van rechtstreeks in de tekst te zoeken, een index doorzoekt. ELK wordt gebruikt door grote bedrijven en andere software zoals Kafka, Netflix en Google. (Elastic, sd)

CloudFlare

CloudFlare wordt gebruikt voor HTTP Caching (opslaan van gegevens zodat je niet telkens hetzelfde op moet halen). (Cloudflare, sd)

Varnish

Varnish maakt gebruik aan het cachen van gegevens. Hierdoor is het mogelijk om de performance flink te verhogen. Dit gaat voor de http-modules zitten en vangt de responses op om ze daarna terug te sturen naar de gebruiker. (Varnish, sd)

Amazon Load balancer

Amazon Load balancer maakt gebruik van geavanceerde routing mogelijkheden om binnenkomende verzoeken naar het vereiste doel te laten navigeren. Amazon Load Balancer is ook zeer geschikt voor Micro-Services en zorgt voor uitstekende beveiliging doordat het garandeert om de nieuwste SSL-protocollen te gebruiken. (Amazon, sd)

Het artikel van Geshan is gebruikt om mogelijke schaalbaarheid tools en processen te raadplegen. (Manandhar, 2020)

Gekozen Schaarbaarheid Tools

De hieronder drie gekozen tools kunnen we A/B testen door te vergelijken met de originele staat van het systeem.

Elastic Search

Kunnen we gebruiken om de laadtijden van de front-end te verlagen.

Varnish

Kunnen we gebruiken om de laadtijden voor de gebruiker aan de front-end zijde te verlagen door gegevens te cachen.

Redis met NoSQL

Kunnen we gebruiken voor de backend kant om de schaalbaarheid van de database te verhogen.

Loadview

Kunnen we gebruiken voor het testen van de reactie van het systeem bij het vermenigvuldigen in het gebruik hiervan. Loadview test wat er gebeurt met het systeem als de load verhoogd wordt.

4. Werkplaats

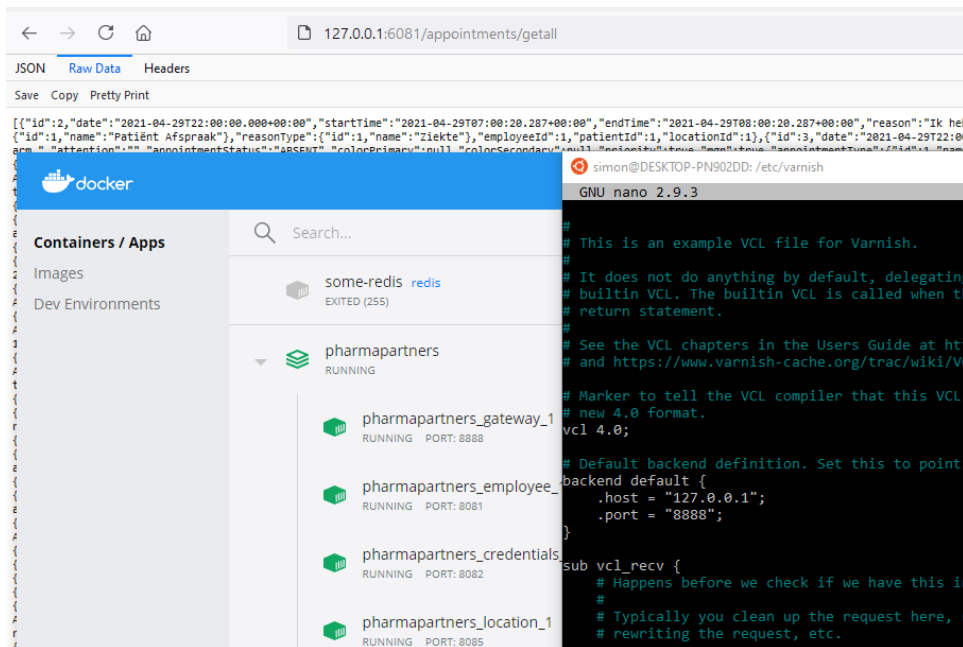
Elastic Search

Een kleine POC met Elastic Search (& Kibana) is opgezet. Hieruit bleek dat er vrijwel alleen monitoring data geanalyseerd kan worden. We hebben dus gekozen om dit niet verder op te zetten omdat dit niet direct nodig is bij het project. Ook zou er te veel tijd in gaan zitten om een POC op te zetten. Dit is omdat er meerdere andere services opgezet zouden moeten worden in een manier waarop ze zouden werken met Elastic Search.

Varnish

Varnish is goed gedocumenteerd op deze website (Varnish Software, sd).

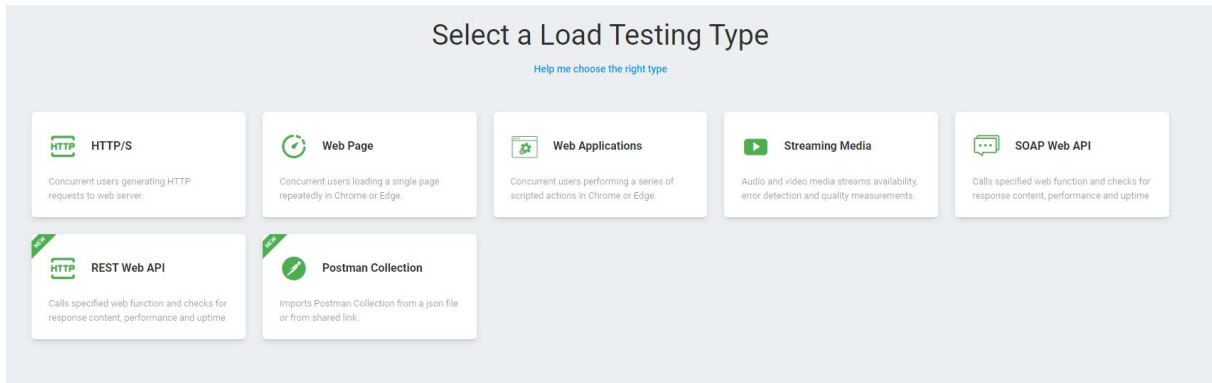
Varnish opgezet met ons profitaak project als host.



Varnish maakt gebruik van Http Caching, dit zorgt ervoor dat een pagina sneller geladen wordt. Door het Cachen van een Http pagina kunnen ook beveiligingsrisico's aan te pas komen. Deze beveiligingsrisico's zijn al bekend binnen OWASP en heeft de naam "Cache Poisoning" gekregen. (Zhong & Rezos, sd)


Loadview

Loadview heeft verschillende opties voor het testen van applicaties als het gaat om de druk die deze applicaties aan kunnen. Dit wordt gedaan door acties uit te voeren die je zelf geconfigureerd hebt. Helaas kunnen de meeste methodes die beschikbaar zijn, alleen gebruikt worden voor het testen van applicaties/ websites die online draaien.





In het programma kun je op basis van een verzoek zoals een Postman opdracht je applicatie testen op de mogelijke drukte van het dagelijkse verkeer. Hier kun je invullen hoeveel gebruikers er beginnen met het uitvoeren van de opdracht, hoeveel hier per minuut bijkomen en voor hoeveel minuten het programma met de bereikte gebruikers door moet gaan met het uitvoeren van de opdracht.

Bij het annuleren van een opdracht worden de berekende kosten nog steeds verrekend. Dit is absurd, want het heeft de server geen resources gekost, omdat de opdracht alvorens het uitvoeren hiervan gestopt was.

Test History: MedicomAgenda							
Run Date, Time	Load Type, Scenario Name	Duration	Users	Errors	State	Cost	Scenario ID
05/28/2021 10:37 AM	 MedicomAge...	03:00	20		CANCELED	\$8.50	55022

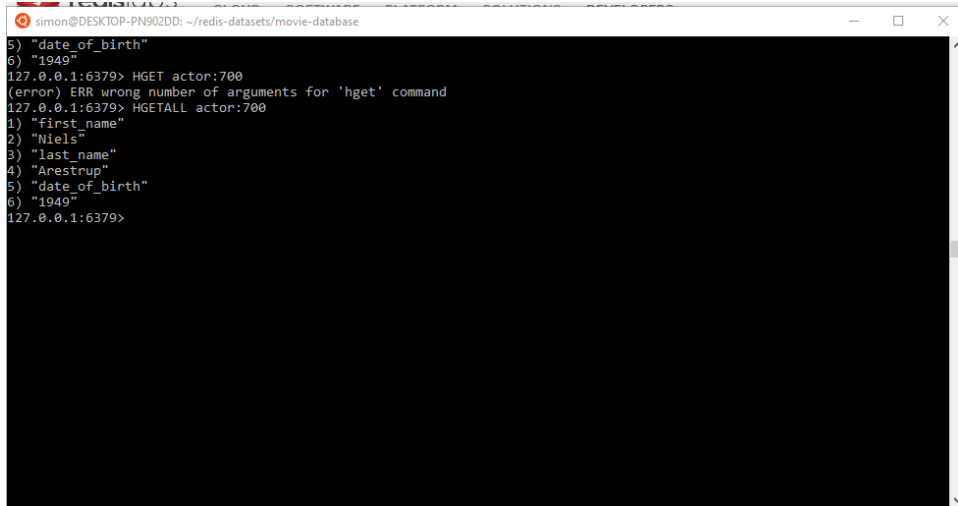
Hieronder is zichtbaar dat de kosten van \$8,50 verrekend zijn. Dit is gelukkig van het gratis tegoed gehaald en dit heeft gezorgd dat ik geen geld hieraan uit heb moeten geven, maar het is nog steeds heel bizar.

Test Manager				+ New Test	
Balance Current Plan: On Demand				 \$11.50 Wallet	
Details					
Tests				Tests: 1	Tasks: 1
				Tasks	Runs
				Last Run	
 MedicomAgenda				1	1
				9 minutes ago	

Redis met NoSQL

We hebben een Redis server opgezet in Ubuntu door gebruikt te maken van WSL (Nielsen, 2018).

Hiernaast is ook een kleine movie database opgezet (tgrall, 2020).



```
simon@DESKTOP-PN902DD: ~/redis-datasets/movie-database
5) "date_of_birth"
6) "1949"
127.0.0.1:6379> HGET actor:700
(error) ERR wrong number of arguments for 'hget' command
127.0.0.1:6379> HGETALL actor:700
1) "first_name"
2) "Niels"
3) "last_name"
4) "AresTrup"
5) "date_of_birth"
6) "1949"
127.0.0.1:6379>
```

De grootte van de test database is 2241 indexes.



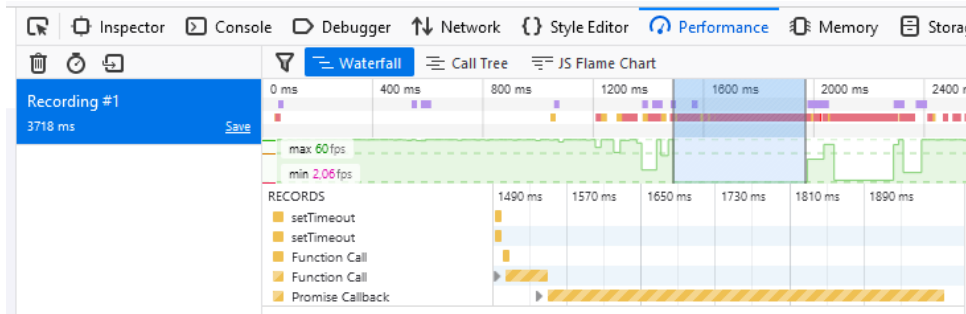
```
simon@DESKTOP-PN902DD: ~/redis-datasets/movie-database
127.0.0.1:6379> DBSIZE
(integer) 2241
127.0.0.1:6379>
```

Dezelfde database heb ik ook in de MSSQL server van fontys neergezet.

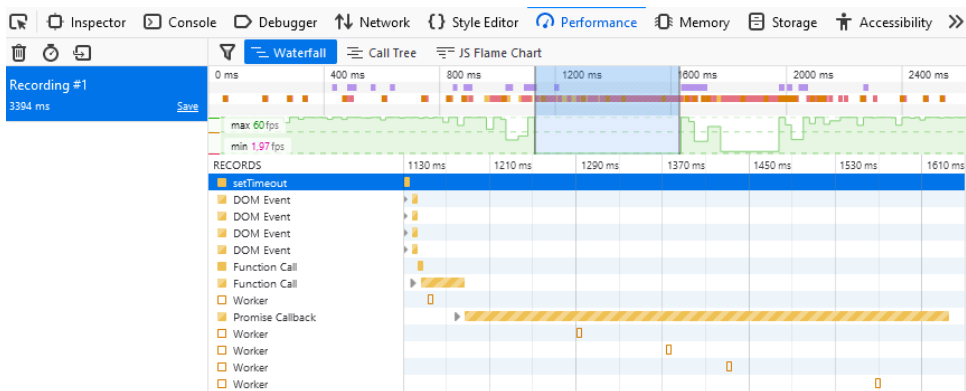
5. Veldwerk

Varnish

Met caching, laadtijd van (1890-1490) 400 ms



Zonder caching, laadtijd van (1610-1130) 480 ms.



Conclusie

De laadtijd met Varnish is 80ms korter. Varnish heeft dus een duidelijke invloed op de laadtijd van de front-end. Het is een klein verschil, maar is mogelijk dat dit verschil groter wordt naarmate er meer gebruikers zich op de website bevinden.

LoadView

Voor de tweede test poging hebben wij gekozen om de port tunnel "NgRok" te gebruiken. Hierdoor zou het in theorie mogelijk zijn voor LoadView om de Front-end die op localhost draait te bereiken.

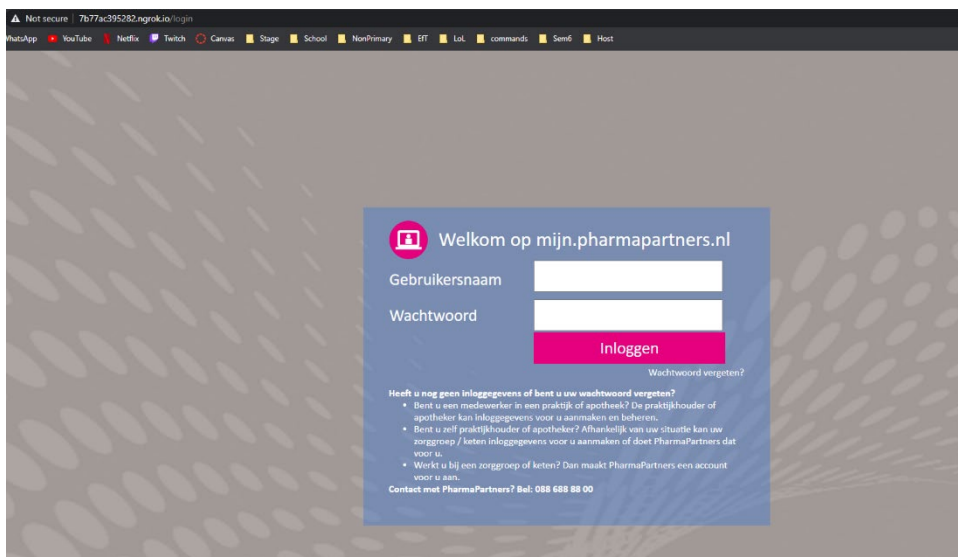
```
C:\Users\LLHVe\Downloads\ngrok.exe - ngrok http ...
ngrok by @inconshreveable (Ctrl+C to quit)

Session Status      online
Session Expires     1 hour, 57 minutes
Version             2.3.40
Region              United States (us)
Web Interface        http://127.0.0.1:4040
Forwarding           http://7b77ac395282.ngrok.io -> http://localhost:4
Forwarding           https://7b77ac395282.ngrok.io -> http://localhost:

Connections          ttl    opn    rt1    rt5    p50    p90
                   36     2     0.27   0.10   0.56   11.53

HTTP Requests
-----
GET /sockjs-node/645/u2ulvbx3/websocket 101 Switching Protocols
GET /sockjs-node/info                    200 OK
GET /sockjs-node/073/nh01zjxb/websocket 101 Switching Protocols
GET /sockjs-node/info                    200 OK
GET /sockjs-node/313/vhffemmo/websocket 101 Switching Protocols
GET /sockjs-node/info                    200 OK
GET /sockjs-node/022/fer3ozr2/websocket 101 Switching Protocols
GET /sockjs-node/info                    200 OK
GET /sockjs-node/138/icjk2p5p/websocket 101 Switching Protocols
```

Het is succesvol gelukt om de Front-End te forwarden via NgRok.




NgRok gaf aan dat we van te veel connecties gebruik hadden gemaakt en dat wanneer we meer connecties per minuut zouden willen aangaan we hiervoor moesten betalen.

Too Many Connections

Too many connections! The tunnel session 'ts_1tNrJq8bd2AeLIlpqneGOghDPma' has violated the rate-limit policy of 20 connections per minute by initiating 37 connections in the last 60 seconds. Please decrease your inbound connection volume or upgrade to a paid plan for additional capacity.


The error encountered was: **ERR_NGROK_702**

In Loadview hebben we de URL opgegeven die door NgRok gegenereerd wordt: <http://7b77ac395282.ngrok.io/agenda/16>.

 URL

Detailed ☐

Enter a URL or IP address. Add or change parameters by editing the Header Fields, Request Type, and GET/POST parameters below.



Ook hebben we in LoadView de Authorization Header toegevoegd die we eerder verkregen door handmatig in te loggen.

▼ Headers

Header Name	Value
Authorization	Bearer eyJhbGciOiJIUzUxMiJ9.eyJzdWiiOiJkb2t0ZXIiLCJleHAiOjE2MjI2MjEzNDcsmIhdCj6MTYyMjYyM

+

 Add Header

Voordat de test begon gaf de validatie aan dat er error te vinden waren. Helaas konden wij niet achterhalen waar de errors daadwerkelijk vandaan kwamen en hebben we

besloten de test alsnog uit te voeren.

Device: <http://7b77ac395282.ngrok.io/agenda/16>

Adjust User Behavior

Edit Device

Task Type	Task Name	URL	Status
HTTPS	http://7b77ac395282.ngrok.io/agenda/16	http://7b77ac395282.ngrok.io/agenda/16	

Task Details

Task Type

HTTPS

Maximum Connec...

60 Sec

FollowRedirects

True

Request Type

GET

URL

<http://7b77ac395282.ngrok.io/agenda/16>

Check Certificate ...

False

Check Certificate ...

False

Check Certificate ...

False

Check Certificate ...

False

Check Certificate ...

False

Full page download

False

Headers

Name	Value
Authorization	Bearer eyJhbGciOiJIUzUxMiJ9.eyJ...

Profile Details

User Behavior

Normal User

Minimum Delay (secs)

3

Maximum Delay (secs)

6

Validating... It may take several minutes to complete, please be patient.

De uiteindelijke test instellingen zagen er als volgt uit.

Device: <http://7b77ac395282.ngrok.io/agenda/16>

Adjust User Behavior

Edit Device

Task Type	Task Name	URL	Status
HTTPS	http://7b77ac395282.ngrok.io/agenda/16	http://7b77ac395282.ngrok.io/agenda/16	1 Error Found

Validation Result: 1 Error found [View Details](#)

Validate

Load Type

Load Step Curve

Load with a pre-determined number of concurrent users for specified time durations.

Goal-Based Curve

Auto adjusts concurrent users to reach a required rate of transactions per time interval.

Dynamic Adjustable Curve

Manually adjust concurrent users in real-time, while the test is running.

Execution Plan

Start with

5

users

Raise by

5

users/min

for

2

min

Hold

15 users

for

1

min

+ Add Action

15

Peak Virtual Users

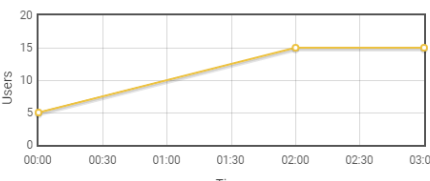
3 min

Test Duration

442

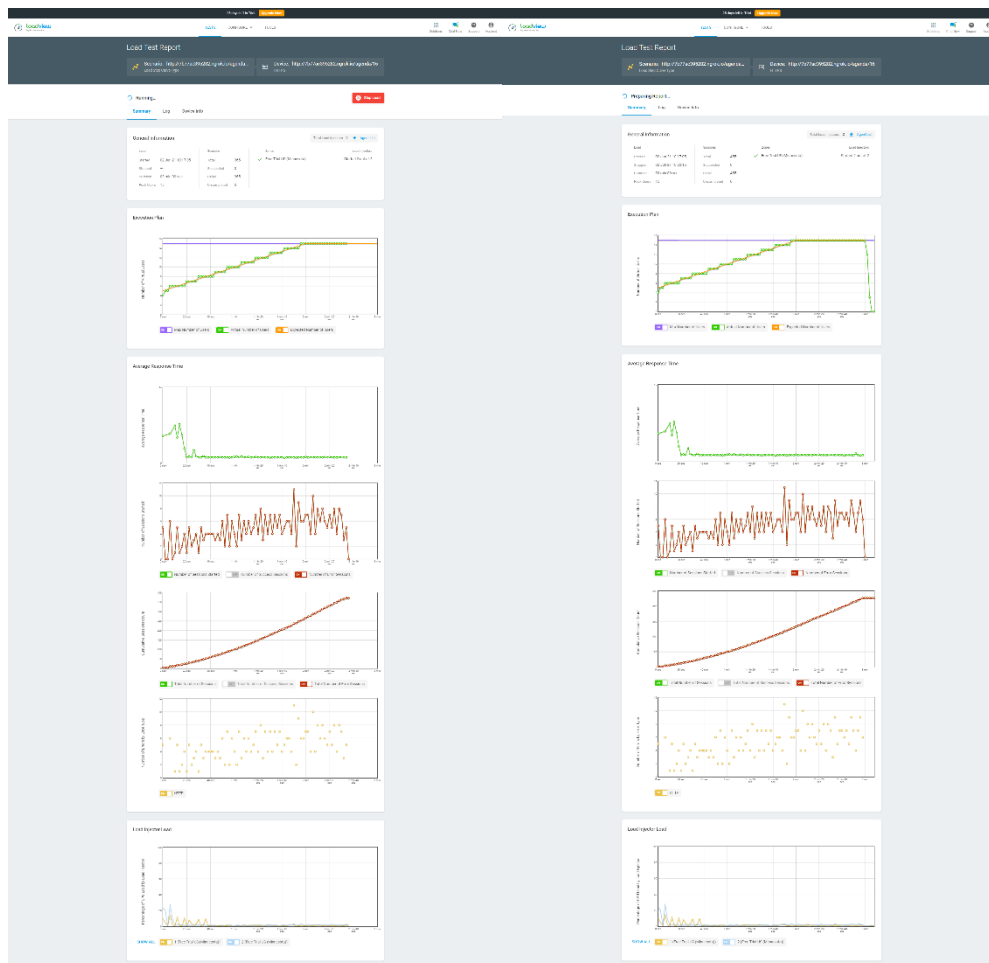
Estimated Sessions

Users



Time

Het uiteindelijke resultaat van de test kwam als geen verrassing. Het gaf namelijk aan dat alle sessies onsuccesvol waren. Dit hadden wij natuurlijk kunnen verwachten vanwege de errors die de validatie van LoadView zelf al vantevoren aangaf.



Conclusie

Loadview geeft de mogelijkheid om te testen indien de architectuur van een applicatie zodanig opgezet is dat er meerdere mensen tegelijkertijd hiervan gebruik kunnen maken. Helaas is Loadview minder toegankelijk voor applicaties die nog niet draaiende zijn op een server. Hierover kan dan ook gezegd wordt dat het alleen nuttig is om Loadview te gebruiken als de architectuur van jouw applicatie al goed opgezet is en je benieuwd bent of deze een stress test volhoudt.

Redis met NoSQL

Het plan om Redis te vergelijken met de MSSQL-server van fontys is om dezelfde dataset en query te timen en het verschil te vergelijken. Mijn verwachtingen zijn dat Redis een stuk sneller is, omdat dit op Ubuntu draait. Verder gebruikt het ook NoSQL dat bekend staat op zijn snelheid.

Omdat NoSQL werkt met key:value en dus geen tabellen met relaties is het vergelijken best lastig. Een simpele getByld is wel te doen door te filteren op key:value.

```
simon@DESKTOP-PN902DD: ~/redis-datasets/movie-database
(empty list or set)
127.0.0.1:6379> MULTI
OK
127.0.0.1:6379> TIME
QUEUED
127.0.0.1:6379> HGETALL actor:500
QUEUED
127.0.0.1:6379> TIME
QUEUED
127.0.0.1:6379> EXEC
1) 1) "1622196391"
   2) "75473"
2) 1) "first_name"
   2) "Nikolaj"
   3) "last_name"
   4) "Coster-Waldau"
   5) "date_of_birth"
   6) "1970"
3) 1) "1622196391"
   2) "75484"
127.0.0.1:6379>
```

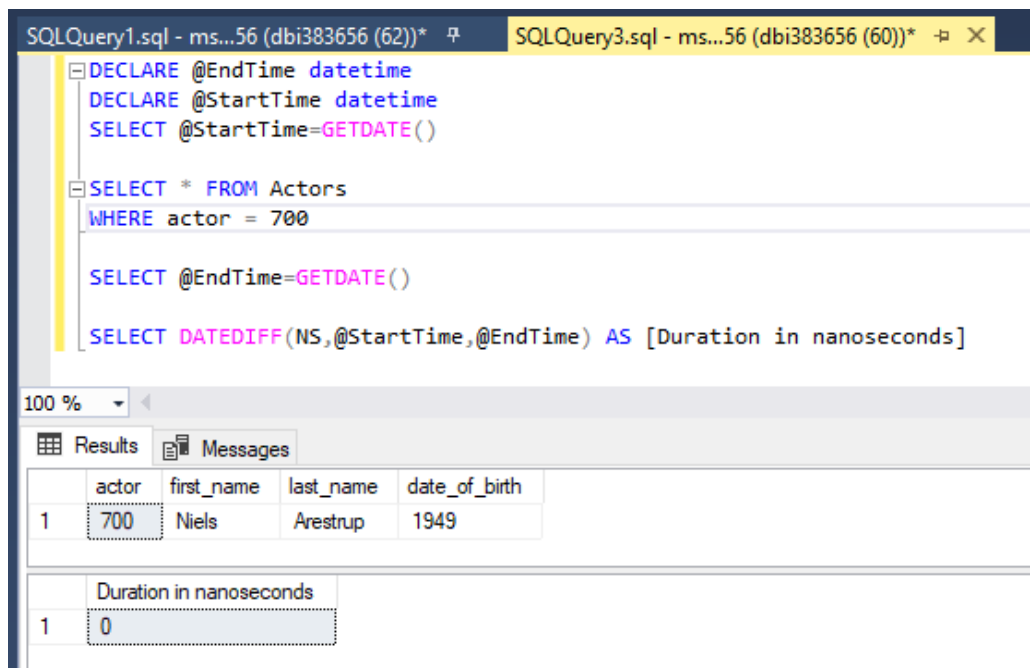
Deze query duurde 11 microseconden.

Een SELECT * FROM actor ziet er zo uit: KEYS 'actor:*

```
127.0.0.1:6379> MULTI
OK
127.0.0.1:6379> TIME
QUEUED
127.0.0.1:6379> KEYS 'actor:*'
QUEUED
127.0.0.1:6379> TIME
QUEUED
127.0.0.1:6379> EXEC
1) 1) "1622201597"
   2) "673897"
2) 1) "actor:373"
   1316) "actor:396"
   1317) "actor:1129"
   1318) "actor:731"
   1319) "actor:1071"
3) 1) "1622201597"
```

Deze query duurde 832 microseconden.

Als we een vergelijkbare query doen in de MSSQL-server van fontys krijgen we dit.



The screenshot shows a SQL Server Enterprise Manager window with two tabs: 'SQLQuery1.sql' and 'SQLQuery3.sql'. The active tab is 'SQLQuery3.sql', which contains the following T-SQL code:

```
DECLARE @EndTime datetime
DECLARE @StartTime datetime
SELECT @StartTime=GETDATE()

SELECT * FROM Actors
WHERE actor = 700

SELECT @EndTime=GETDATE()




SELECT DATEDIFF(NS,@StartTime,@EndTime) AS [Duration in nanoseconds]
```

Below the query editor, the 'Results' tab is selected, displaying two tables. The first table, 'Actors', has columns 'actor', 'first_name', 'last_name', and 'date_of_birth'. The second table, 'Duration in nanoseconds', has columns 'Duration in nanoseconds'.

	actor	first_name	last_name	date_of_birth
1	700	Niels	Arestrup	1949

	Duration in nanoseconds
1	0

Ik kan niet geloven dat de query écht maar 0 nanoseconden duurt om uit te voeren, dus ik verwacht dat de MSSQL-Server van Fontys maar om de zoveel tijd de datum aan past. Als ik een SELECT * query uit voer dan zit de lengte zog steeds op 0 nanoseconden of springt deze naar 3000000 nanoseconden.

SQLQuery1.sql - ms...56 (dbi383656 (62))*  SQLQuery3.sql - ms...56 (dbi383656 (60))*  

```


DECLARE @EndTime datetime
DECLARE @StartTime datetime
SELECT @StartTime=GETDATE()



SELECT * FROM Actors

SELECT @EndTime=GETDATE()

SELECT DATEDIFF(NS,@StartTime,@EndTime) AS [Duration in nanoseconds]




```

100 % 

 Results  Messages

	actor	first_name	last_name	date_of_birth
1	1	Chris	Pratt	1979
2	2	Zoe	Saldana	1978
3	3	Dave	Bautista	1969
4	4	Vin	Diesel	1967
5	5	Bradley	Cooper	1975

	Duration in nanoseconds
1	0

SQLQuery1.sql - ms...56 (dbi383656 (62))*  SQLQuery3.sql - ms...56 (dbi383656 (64))*  

```


DECLARE @EndTime datetime
DECLARE @StartTime datetime
SELECT @StartTime=GETDATE()



SELECT * FROM Actors

SELECT @EndTime=GETDATE()

SELECT DATEDIFF(NS,@StartTime,@EndTime) AS [Duration in nanoseconds]

```

100 % 

 Results  Messages

	actor	first_name	last_name	date_of_birth
1	1	Chris	Pratt	1979
2	2	Zoe	Saldana	1978
3	3	Dave	Bautista	1969
4	4	Vin	Diesel	1967
5	5	Bradley	Cooper	1975
6	6	Lee	Pace	1979

	Duration in nanoseconds
1	3000000

Conclusie

Uit deze test is dus niet te zien welke van de twee methoden sneller is. Wel is het te zien dat een query makkelijker te timen is in Redis dan SQL. Verder is Redis dus ook meer accuraat dan SQL.

6. Showroom

Hieronder is een link naar de presentatie weergegeven die we gebruiken hebben voor de pitch van de resultaten van het onderzoek.

https://stichtingfontys-my.sharepoint.com/:p:/g/personal/386215_student_fontys_nl/ER5ojGAuMINMoJZHV5eTNDUBdrIN270JSPrsxCbcHv-k8Q?e=Ua1TcJ

Advies

Wij hebben een aantal conclusies kunnen trekken uit ons onderzoek. Op basis van deze conclusies geven wij de volgende adviezen.

Voor het verminderen van de druk op de front-end wanneer veel gebruikers zich hierop bevinden, raden wij aan om gebruik te maken van Varnish. Uit ons onderzoek bleek dat door Varnish te gebruiken er een vermindering in de laadtijd plaatsvindt. Dit is echter alleen op een kleine schaal getest en de vermindering in de laadtijd kan verschillen wanneer Varnish voor een grotere applicatie gebruikt wordt.

Voor het stresstesten van laadtijden en de architectuur van applicaties kan er gebruik gemaakt worden van Loadview. De configuratie en mogelijkheden die Loadview aan de gebruiker aanbiedt zijn doordacht op gebruikersvriendelijkheid. Het uitvoeren van de stresstesten gebeurt op een gestructureerde manier, waardoor de gebruiker direct zijn weet wat hem te wachten staat. Tegenover deze functionaliteiten staat echter een fikse kostenplaatje wat niet iedereen wil betalen voor deze diensten. Door de hiervoor genoemde redenen, adviseren wij aan het bedrijf om zelf een beslissing te nemen als het gaat over het gebruiken van deze dienst, vanwege de hoge kosten die hieraan verbonden zitten.

Het is niet te concluderen uit ons onderzoek dat NoSQL in combinatie met Redis een verkortere laadtijd geeft in vergelijking met MSSQL. Dit doordat de weergegeven tijdsduur van een Query in MSSQL moeilijker te bepalen is in vergelijking met het bepalen van de tijdsduur in NoSQL. Vanwege de niet bepalende uitslag van ons onderzoek raden wij aan om gebruik te maken van de SQL-variant waar de eigen voorkeur naar uitgaat. Verder is het belangrijk om te vermelden dat het vergelijken van NoSQL met MSSQL gezien kan worden als het vergelijken van appels met peren.

Literatuurlijst

- Amazon. (n.d.). *Application Load Balancer / Elastic Load Balancing / Amazon Web Services*. Retrieved from [aws.amazon.com](https://aws.amazon.com/elasticloadbalancing/application-load-balancer/):
<https://aws.amazon.com/elasticloadbalancing/application-load-balancer/>
- Apache Solr. (n.d.). *Features - Apache Solr*. Retrieved from [solr.apache.org](https://solr.apache.org/features.html#solrcloud):
<https://solr.apache.org/features.html#solrcloud>
- Cloudflare. (n.d.). *Cloudflare - The Web Performance & Security Company / Cloudflare*. Retrieved from www.cloudflare.com: <https://www.cloudflare.com/>
- Elastic. (n.d.). *Elastic Enterprise Search / Elastic*. Retrieved from [www.elastic.co](https://www.elastic.co/enterprise-search):
<https://www.elastic.co/enterprise-search>
- Kubernetes. (n.d.). *What is Kubernetes? / Kubernetes*. Retrieved from [kubernetes.io](https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/):
<https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/>
- LoadView. (n.d.). *Load Testing & Website Performance Tools - LoadView*. Retrieved from www.loadview-testing.com: <https://www.loadview-testing.com/>
- Manandhar, G. (2020, december 7). *Software scalability: 6 dev and operations factors to consider for meeting high demands*. Retrieved from [geshan.com.np](https://geshan.com.np/blog/2020/12/software-scalability/):
<https://geshan.com.np/blog/2020/12/software-scalability/>
- MongoDB. (n.d.). *What is NoSQL? NoSQL Databases Explained / MongoDB*. Retrieved from [www.mongodb.com](https://www.mongodb.com/nosql-explained): <https://www.mongodb.com/nosql-explained>
- Nielsen, D. (2018, juli 27). *Running Redis on Windows 10 / Redis Labs*. Retrieved from [redislabs.com](https://redislabs.com/blog/redis-on-windows-10/): <https://redislabs.com/blog/redis-on-windows-10/>
- Qlik. (n.d.). *ds-technical-brief-qlik-scalability-tools-en.pdf*. Retrieved from [www.qlik.com](https://www.qlik.com/us/-/media/files/resource-library/global-us/direct/datasheets/ds-technical-brief-qlik-scalability-tools-en.pdf):
<https://www.qlik.com/us/-/media/files/resource-library/global-us/direct/datasheets/ds-technical-brief-qlik-scalability-tools-en.pdf>
- Redis. (n.d.). *Introduction to Redis – Redis*. Retrieved from [redis.io](https://redis.io/topics/introduction):
<https://redis.io/topics/introduction>
- tgrall. (2020, september 16). *redis-datasets/movie-database at master · redis-developer/redis-datasets*. Retrieved from [github.com](https://github.com/redis-developer/redis-datasets/tree/master/movie-database): <https://github.com/redis-developer/redis-datasets/tree/master/movie-database>
- Varnish. (n.d.). *Introduction to Varnish – Varnish HTTP Cache*. Retrieved from [varnish-cache.org](https://varnish-cache.org/intro/index.html#intro): <https://varnish-cache.org/intro/index.html#intro>
- Varnish Software. (n.d.). *Varnish Installation – Varnish version trunk documentation*. Retrieved from [varnish-cache.org](https://varnish-cache.org/docs/trunk/installation/index.html): <https://varnish-cache.org/docs/trunk/installation/index.html>
- Zhong, W., & Rezos. (n.d.). *Cache Poisoning Software Attack / OWASP Foundation*. Retrieved from [owasp.org](https://owasp.org/www-community/attacks/Cache_Poisoning): https://owasp.org/www-community/attacks/Cache_Poisoning

ZooKeeper. (n.d.). *ZooKeeper: Because Coordinating Distributed Systems is a Zoo*.
Retrieved from [zookeeper.apache.org](https://zookeeper.apache.org/doc/current/zookeeperOver.html):
<https://zookeeper.apache.org/doc/current/zookeeperOver.html>