

# Automatisch uitrol van SQL systemen en vergelijking van beschikbaarheid

Thomas Uyttendaele

Thesis voorgedragen tot het behalen  
van de graad van Master of Science  
in de ingenieurswetenschappen:  
computerwetenschappen,  
hoofdspecialisatie Gedistribueerde  
systemen

**Promotor:**

Prof. dr. ir. Wouter Joosen

**Assessor:**

Ir.

**Begeleider:**

Ir. B. Vanbrabant

© Copyright KU Leuven

Zonder voorafgaande schriftelijke toestemming van zowel de promotor als de auteur is overnemen, kopiëren, gebruiken of realiseren van deze uitgave of gedeelten ervan verboden. Voor aanvragen tot of informatie i.v.m. het overnemen en/of gebruik en/of realisatie van gedeelten uit deze publicatie, wend u tot het Departement Computerwetenschappen, Celestijnenlaan 200A bus 2402, B-3001 Heverlee, +32-16-327700 of via e-mail [info@cs.kuleuven.be](mailto:info@cs.kuleuven.be).

Voorafgaande schriftelijke toestemming van de promotor is eveneens vereist voor het aanwenden van de in deze masterproef beschreven (originele) methoden, producten, schakelingen en programma's voor industrieel of commercieel nut en voor de inzending van deze publicatie ter deelname aan wetenschappelijke prijzen of wedstrijden.

# Voorwoord

Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetur eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

Etiam euismod. Fusce facilisis lacinia dui. Suspendisse potenti. In mi erat, cursus id, nonummy sed, ullamcorper eget, sapien. Praesent pretium, magna in eleifend egestas, pede pede pretium lorem, quis consectetur tortor sapien facilisis magna. Mauris quis magna varius nulla scelerisque imperdiet. Aliquam non quam. Aliquam porttitor quam a lacus. Praesent vel arcu ut tortor cursus volutpat. In vitae pede quis diam bibendum placerat. Fusce elementum convallis neque. Sed dolor orci, scelerisque ac, dapibus nec, ultricies ut, mi. Duis nec dui quis leo sagittis commodo.

Aliquam lectus. Vivamus leo. Quisque ornare tellus ullamcorper nulla. Mauris porttitor pharetra tortor. Sed fringilla justo sed mauris. Mauris tellus. Sed non leo. Nullam elementum, magna in cursus sodales, augue est scelerisque sapien, venenatis congue nulla arcu et pede. Ut suscipit enim vel sapien. Donec congue. Maecenas urna mi, suscipit in, placerat ut, vestibulum ut, massa. Fusce ultrices nulla et nisl.

Voorwoord  
schrijven

*Thomas Uyttendaele*

# Inhoudsopgave

<b>Voorwoord</b>	<b>i</b>
<b>Samenvatting</b>	<b>iv</b>
<b>Lijst van figuren en tabellen</b>	<b>v</b>
<b>Lijst van afkortingen en symbolen</b>	<b>vi</b>
<b>1 Inleiding</b>	<b>1</b>
<b>2 Probleem- en doelstelling</b>	<b>3</b>
2.1 Probleemstelling . . . . .	3
2.2 Doelstelling . . . . .	6
<b>3 Selectie van databases</b>	<b>7</b>
3.1 Column . . . . .	7
3.2 Key- . . . . .	7
<b>4 Automatisch uitrollen van databases</b>	<b>9</b>
4.1 Inleiding . . . . .	9
4.2 IMP: Framework voor installatie . . . . .	9
<b>5 Besluit</b>	<b>11</b>
<b>Bibliografie</b>	<b>16</b>

# Todo list

Voorwoord schrijven . . . . .	i
Abstract schrijven . . . . .	iv
Schrijf inleiding . . . . .	1

# Samenvatting

Abstract schrijven

In dit **abstract** environment wordt een al dan niet uitgebreide samenvatting van het werk gegeven. De bedoeling is wel dat dit tot 1 bladzijde beperkt blijft. Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetur eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

Etiam euismod. Fusce facilisis lacinia dui. Suspendisse potenti. In mi erat, cursus id, nonummy sed, ullamcorper eget, sapien. Praesent pretium, magna in eleifend egestas, pede pede pretium lorem, quis consectetur tortor sapien facilisis magna. Mauris quis magna varius nulla scelerisque imperdiet. Aliquam non quam. Aliquam porttitor quam a lacus. Praesent vel arcu ut tortor cursus volutpat. In vitae pede quis diam bibendum placerat. Fusce elementum convallis neque. Sed dolor orci, scelerisque ac, dapibus nec, ultricies ut, mi. Duis nec dui quis leo sagittis commodo.

Aliquam lectus. Vivamus leo. Quisque ornare tellus ullamcorper nulla. Mauris porttitor pharetra tortor. Sed fringilla justo sed mauris. Mauris tellus. Sed non leo. Nullam elementum, magna in cursus sodales, augue est scelerisque sapien, venenatis congue nulla arcu et pede. Ut suscipit enim vel sapien. Donec congue. Maecenas urna mi, suscipit in, placerat ut, vestibulum ut, massa. Fusce ultrices nulla et nisl.

# Lijst van figuren en tabellen

Lijst van figuren

Lijst van tabellen

3.1 Classificatie systeem . . . . .	7
-------------------------------------	---

# Lijst van afkortingen en symbolen

## Afkortingen

IMP      Infrastructure Management Platform

## Symbolen

42      aaa



# Hoofdstuk 1

## Inleiding

In dit hoofdstuk wordt het werk ingeleid. Het doel wordt gedefinieerd en er wordt uitgelegd wat de te volgen weg is (beter bekend als de rode draad).

Schrijf inleiding



# Hoofdstuk 2

## Probleem- en doelstelling

### 2.1 Probleemstelling

Sinds het begin van WEB 2.0 is het gebruik van het internet veranderd, van een statisch tot een dynamische omgeving waar iedereen een persoonlijke en individuele ervaring heeft. Samen met deze evolutie is ook het type, de hoeveelheid en het gebruik van data en de opslag ervan veranderd.

Deze verandering kan allereerst toegeschreven worden aan de toename in het aantal website, daarnaast is ook de content van een webpagina veranderd, geen statische pagina's met enkel tekst maar ook afbeeldingen en (korte) videofragmenten zijn aanwezig. Deze content wordt ook niet meer enkel door de websitebeheerder online geplaatst maar de gebruikers kunnen zelf hun data invoegen en krijgen gepersonaliseerde en dynamische pagina's te zien. Tenslotte is ook de toename in het aantal gebruikers, op 30 juni 2012 waren er meer als 2.4 miljard gebruikers van het internet, een toename van meer als 500% op 12 jaar tijd. [3]

Al deze veranderingen hebben ook hun doorslag op de achterliggende infrastructuur. Waar kleinere en statische websites nog konden gehost worden op een enkele server, is dit niet meer het geval voor hedendaagse (populaire) websites. Een onderdeel van deze web applicaties is de databases waarin de vereiste data kan in worden opgeslagen worden en later opgevraagd worden. Veel van deze databases zijn ontworpen naar het systeem van de relationele databases, uitgewerkt naar Codd [2].

Voor talrijke van de huidige applicaties is de voorwaarden voor dataopslag verschillend met deze van de jaren 70. Enkele systemen hebben geen nood meer aan onmiddellijke consistentie maar hebben genoeg aan **eventuele consistentie**. Onder andere omwille van deze reden is er een stroming van nieuwe database systemen gekomen genaamd **NoSQL**. Onder deze noemer vallen vele systemen, elk eigen in hun soort en aangepast aan bepaalde manieren van dataopslag en de daarvoor nodige vereisten, een belangrijk element is het **horizontaal schaalbaar** zijn van deze systemen. Met andere woorden, deze databases zijn ontworpen om de data over meerdere systemen te verspreiden om de load te verspreiden.

**Horizontaal schaalbaar** brengt verschillende voordelen mee, maar zorgt ook voor 2 problemen die in deze thesis zullen aanbod komen, namelijk de uitrol en daarnaast wat de implicaties zijn van een gedistribueerd datasysteem.

### 2.1.1 Uitrollen van gedistribueerde databases

Voor geavanceerde toepassingen is het gebruik van gespecialiseerde en/of gedistribueerde databases een noodzaak. Maar de tijdsduur van de beslissing dat dit het geval is tot een werkende omgeving hebben, is niet te onderschatten en dit omwille van verschillende componenten.

Een eerste stap die doorlopen wordt, is het opzoeken van de nodige informatie over de verschillende databases, nu bestaan er al verschillende websites en papers waar informatie opgezocht kan worden, maar een consistente test bestaat niet. Een voorbeeld van een vergelijkingswebsite is bijvoorbeeld vsChart [4], ook verschillende papers brengen vergelijkingen naar voor in onder andere performantie en de verschillende data modellen. Maar in de meeste gevallen blijft het aangeraden om de informatie op te zoeken op de website van de software zelf, deze evolueert snel en informatie van een jaar oud kan al verouderd zijn en zo incorrect.

Al deze informatie zou de ontwikkelaar een mogelijkheid geven om de verschillende systemen te kunnen vergelijken en een keuze te maken met welke database hij verder wenst te gaan.

Na de selectie van een specifieke database, komt de keuze voor de opstelling van de database. Bepaalde databases hebben ondersteuning voor Master-slave, andere hebben ondersteuning voor sharding en nog andere hebben een combinatie van de twee. Deze verschillende manieren zorgt ervoor dat er geen eenduidige manier is om alle verschillende systemen consequent uit te rollen. De ontwikkelaar zal in deze stap een opstellen kiezen onder welke hij de database zal uitrollen.

Nu de databasesoftware en de opstelling is gekozen, kunnen de verschillende machines fysiek opgesteld worden. Dit wil zeggen, de opstelling wordt fysiek opgesteld: de computers kiezen, deze op de juiste locatie zetten en te verbinden. De meeste **NoSQL** systemen zijn ontworpen om te draaien op commodity hardware wat wil zeggen dat gewone consumenten computers volstaan.

In het volgend stadium, komt de softwarematige installatie: beginnende bij het besturingssysteem tot de database software zelf. De meeste van deze databases draaien onder verschillende Linux versies. De meeste hebben wel de nodige andere software nodig om ook te kunnen draaien, een voorbeeld is HBase dat Java en Hadoop nodig heeft.

Deze systemen worden geïnstalleerd en nadien geconfigureerd. Waar voor de installatie meestal nog standaard tools zoals **yum** of **apt-get** gebruikt kunnen worden, is de configuratie afhankelijk van systeem tot systeem. Bepaalde systemen zoals

Pgpool werken volledig met configuratie bestanden, andere systemen zoals MongoDB werken voornamelijk met een configuratie via de shell, nog andere, zoals HBase, hebben een basis configuratie waarbij men zich aanmeldt bij een node en daar de rest van de configuratie ontvangen. Deze verschillende aanpakken maken het opzetten van verschillende systemen niet eenvoudiger.

Nu het systeem draait, kan het systeem getest worden en bekeken worden of het voldoet aan de vooropgesteld benodigdheden. Indien dit niet het geval is, kan een aanpassing aan de opstelling misschien voldoen of dient een ander systeem geselecteerd worden.

Vervolgens kan de ontwerpen verder gaan met het ontwerpen van de applicatie, waar de databaselaag uiteindelijk maar een gedeelte van is.

Daarnaast zou het kunnen dat de set-up voldoet tijdens het ontwerpen maar niet tijdens de uiteindelijke ingebruikname of maakt de applicatie een groei mee waardoor er extra resources nodig zijn. Nu zijn veel van de databases en zeker de **NoSQL** databases geschreven om eenvoudig **horizontaal schaalbaar** te zijn: het toevoegen of verwijderen van systemen kan in veel gevallen zelfs dynamisch.

Deze stappen zijn allemaal niet zo onlogisch maar vragen wel tijd en kennis van de installatie en configuratie van deze systemen. Bij veel lokale systemen wordt de installatie en configuratie automatisch gedaan door onder andere gebruik te maken van **yum** of **apt-get** waarna deze systemen volledig werken, want uiteindelijk dient een ontwerper om nieuwe systemen te ontwikkelen, oa. door het samenbrengen van verschillende componenten die als een black block kunnen beschouwd worden met zo weinig mogelijk configuratie.

### 2.1.2 Implicatie van gedistribueerde datasystemen

Een tweede element van de thesis is het onderzoek naar de implicaties van gedistribueerde datasystemen. In dit gebied is er al veel onderzoek gedaan naar de verschillen in datamodellen en performantie maar er is nog geen gerelateerd onderzoek naar de implicaties op de beschikbaarheid van de data.

De CAP-theorie van E. Brewer[1] staat voor consistency, availability en partition tolerance. De volgende definities zoals geformuleerd in [5]:

**Consistency** of consistentie betekent dat na een aanpassing van een schrijver in een gedeelte databron, alle lezers deze aanpassing zien.

**Availability** of beschikbaarheid, meer bepaald hoge beschikbaarheid, betekent dat een systeem de mogelijkheid heeft om verdere operaties (lezen en/of schrijven) te verwerken indien een of meerdere nodes in de cluster niet beschikbaar zijn.

**Partition tolerance** of partitie tolerantie betekent dat het systeem verder kan blijven werken in zijn geheel indien de volledige cluster (tijdelijke of finaal) in twee of meer delen opgedeeld is. Een andere betekenis is dat er de mogelijkheid is tot het dynamisch verwijderen of toevoegen van nodes, waarbij de te verwijderen of toe te voegen nodes gezien worden als een eigen netwerk partitie.

Vervolgens stelt E. Brewer dat in een systeem met gedeelde data maximaal aan 2 van de 3 elementen van de CAP-theorie kan voldaan worden.

Dit heeft invloed op de beschikbaarheid van de data in gedistribueerde databases, maar verschillende database systemen kiezen verschillende beschikbaarheden. Bepaalde database systemen geven geen consistentie maar enkel eventuele consistentie waarbij de aanpassing na verloop van tijd maar beschikbaar is voor alle gebruikers. Indien een ontwerper hier geen rekening bij houdt bij de implementatie, kunnen in reservatiesysteem bijvoorbeeld dubbele boekingen voorkomen over de tijd.

## 2.2 Doelstelling

De doelstellingen voor deze thesis bestaat uit 2 delen, net zoals de probleemstelling. In eerste instantie de uitrol van gedistribueerde databases, in tweede instantie de implicaties op de beschikbaarheid van de data.

### 2.2.1 Uitrol van gedistribueerde databases

Het eerste gedeelte van de doelstellingen is gerelateerd met de uitrol van gedistribueerde databases, voornamelijk het vereenvoudigen van de uitrol.

Net zoals `yum` en `apt-get` de mogelijkheid geven om lokaal op een eenvoudige manier software te installeren, zal in deze thesis de uitrol van verschillende database systemen geautomatiseerd worden en dit door middel van `IDP`.

Aangezien er geen unieke opstelling is, zal er nog steeds een configuratie nodig zijn. Maar het doel is om deze configuratie eenvoudiger te maken op basis van een model.

De gebruiker zal enkel een systeem moeten selecteren, de set-up bepalen en de systemen fysiek opstellen. De softwarematige installatie en configuratie zal niet langer door gebruiker moeten gebeuren maar bepaalde componenten toewijzen aan bepaalde machines, met de nodige relaties tussen de componenten.

### 2.2.2 Implicatie op beschikbaarheid van gedistribueerde datasystemen

Naast de automatisatie van de uitrol, zal er ook gefocust worden op het testen van deze systemen. Dit niet op basis van de performantie maar dit op de beschikbaarheid van de data.

Zoals vermeldt in de probleemstelling, hebben verschillende database systemen een ander methode om om te gaan met het niet beschikbaar zijn van een node en netwerk partities.

Om deze verschillen te vergelijken zullen er testen geschreven worden om de keuzes en hun implicaties duidelijk te maken.

Tenslotte zullen deze testen ook beschikbaar gesteld worden om op een eenvoudige manier ook op andere systemen uitgerold te worden en verificatie van de data mogelijk te maken. Daarnaast is het zo ook mogelijk om deze testen opnieuw uit te voeren indien er updates beschikbaar komen, die mogelijke verbeteringen brengen.

## Hoofdstuk 3

# Selectie van databases

Soort	Performantie	Schaalbaarheid	Flexibiliteit	Complexiteit	Functionaliteit
Key-Value	hoog	hoog	hoog	geen	variabel (geen)
Column	hoog	hoog	gematigd	laag	minimaal
Document	hoog	variabel(hoog)	hoog	laag	variabel (laag)
Graph	variabel	variabel	hoog	hoog	graph theory
Relational	variabel	variabel	laag	gematigd	relational algebra

TABEL 3.1: Classificatie systeem

### 3.1 Column

### 3.2 Key-





## Hoofdstuk 4

# Automatisch uitrollen van databases

### 4.1 Inleiding

### 4.2 IMP: Framework voor installatie



## Hoofdstuk 5

# Besluit

De masterproeftekst wordt afgesloten met een hoofdstuk waarin alle besluiten nog eens samengevat worden. Dit is ook de plaats voor suggesties naar het verder gebruik van de resultaten, zowel industriële toepassingen als verder onderzoek.



# Bijlagen





# Bibliografie

- [1] Eric A. Brewer. „Towards Robust Distributed Systems (Abstract)”. In: *Proceedings of the Nineteenth Annual ACM Symposium on Principles of Distributed Computing*. PODC '00. Portland, Oregon, USA: ACM, 2000, p. 7–. ISBN: 1-58113-183-6. DOI: [10.1145/343477.343502](https://doi.org/10.1145/343477.343502). URL: <http://doi.acm.org/10.1145/343477.343502>.
- [2] E. F. Codd. „A Relational Model of Data for Large Shared Data Banks”. In: *Commun. ACM* 13.6 (jun 1970), p. 377–387. ISSN: 0001-0782. DOI: [10.1145/362384.362685](https://doi.org/10.1145/362384.362685). URL: <http://doi.acm.org/10.1145/362384.362685>.
- [3] Miniwatts Marketing Group. *World Internet Users Statistics Usage and World Population Stats*. Dec 2013. URL: <http://www.internetworldstats.com/stats.htm> (bezocht op 14-12-2013).
- [4] Florian Heinze. *vsChart.com – The Comparison Wiki*. Dec 2013. URL: <http://vschart.com/> (bezocht op 14-12-2013).
- [5] Christof Strauch. *NoSQL Databases*. 2010. URL: <http://www.christof-strauch.de/nosql dbs.pdf>.



## Fiche masterproef

*Student:* Thomas Uyttendaele

*Titel:* Automatisch uitrol van SQL systemen en vergelijking van beschikbaarheid

*Engelse titel:* Automatisch uitrol van SQL systemen en vergelijking van beschikbaarheid

*UDC:* 621.3

*Korte inhoud:*

Hier komt een heel bondig abstract van hooguit 500 woorden.  $\text{\LaTeX}$  commando's mogen hier gebruikt worden. Blanco lijnen (of het commando `\par`) zijn wel niet toegelaten!

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Thesis voorgedragen tot het behalen van de graad van Master of Science in de ingenieurswetenschappen: computerwetenschappen, hoofdspecialisatie  
Gedistribueerde systemen

*Promotor:* Prof. dr. ir. Wouter Joosen

*Assessor:* Ir.

*Begeleider:* Ir. B. Vanbrabant