

# Configuratieafhankelijkheden gebruiken om gedistribueerde applicaties efficiënt te beheren in een hybride cloud

Harm De Weirdt

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

**Assessoren:**

Ir. W. Eetveel

W. Eetrest

**Begeleider:**

Ir. Bart 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

Dit is mijn dankwoord om iedereen te danken die mij bezig gehouden heeft. Hierbij dank ik mijn promotor, mijn begeleider en de voltallige jury. Ook mijn familie heeft mij erg gesteund natuurlijk.

*Harm De Weirdt*

# Inhoudsopgave

<b>Voorwoord</b>	<b>i</b>
<b>Samenvatting</b>	<b>iii</b>
<b>Lijst van figuren en tabellen</b>	<b>iv</b>
<b>1 Inleiding</b>	<b>1</b>
<b>2 Analyse van het probleem</b>	<b>5</b>
2.1 Soorten afhankelijkheden . . . . .	5
2.2 Besluit van dit hoofdstuk . . . . .	6
<b>3 Evaluatie</b>	<b>7</b>
3.1 Besluit van dit hoofdstuk . . . . .	7
<b>4 Besluit</b>	<b>9</b>
<b>A De eerste bijlage</b>	<b>13</b>
A.1 Meer lorem . . . . .	13
A.2 Lorem 51 . . . . .	13

# Samenvatting

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.

# Lijst van figuren en tabellen

Lijst van figuren

Lijst van tabellen

# Hoofdstuk 1

## Inleiding

Configuratiebeheergereedschappen zijn ontwikkeld om het leven van systeembeheerders makkelijker te maken. De serverinfrastructuren die ze moeten onderhouden worden steeds uitgebreider en complexer. Manueel elke server configureren kost niet alleen teveel tijd maar is ook erg foutgevoelig. Het gebruik van scripts is al een stap in de goede richting maar is nog steeds niet voldoende. Als bijvoorbeeld ssh gebruikt wordt om een reeks servers up te daten en één ervan is niet beschikbaar is er plots een verschil tussen systemen die eigenlijk dezelfde configuratie zouden moeten hebben. Een andere manier om een verzameling gelijkaardige machines van hun initiële configuratie te voorzien is het gebruik van images. Daarbij wordt eerst één machine manueel geconfigureerd en daarna de volledige set-up gekloond naar de rest van de servers. Deze methode werkt niet meer voor het verdere onderhoud van de configuraties.

Herschrijven

Dit probleem komt nog prominenter voor als de infrastructuur niet lokaal maar in de cloud gehost wordt. Een groot voordeel van werken in de cloud is de flexibiliteit waarmee servers kunnen toegevoegd en weggenomen kunnen worden. Dit proces gebeurt vaak zelfs automatisch waardoor manuele configuratie helemaal geen optie meer is. In een dergelijke omgeving is een tool die uit zichzelf de volledige infrastructuur kan beheren bijna een noodzaak. Configuratiebeheergereedschappen (of CMS: Configuration Management Software) zoals IMP, Puppet, CFEngine,... laten toe op een efficiënte manier IT infrastructuren op te zetten en onderhouden.

source?

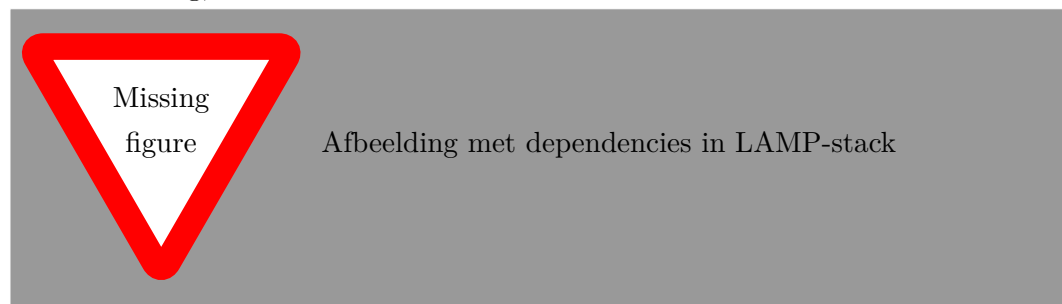
De gebruiker van een dergelijke tool specificeert eerst een model dat de gewenste toestand van de volledige infrastructuur beschrijft. Dit model bestaat uit een oplijsting van machines met de gewenste aanwezige resources (bestanden, mappen, services,...) die ze moeten aanbieden. Een eenvoudige voorbeeldconfiguratie is de LAMP-stack:

Referenties naar sites van hoofdpaginas van CMSen

- Linux
- Apache webserver
- MySQL database (of MariaDB, MongoDB,...)
- PHP (of Perl, Python,...)

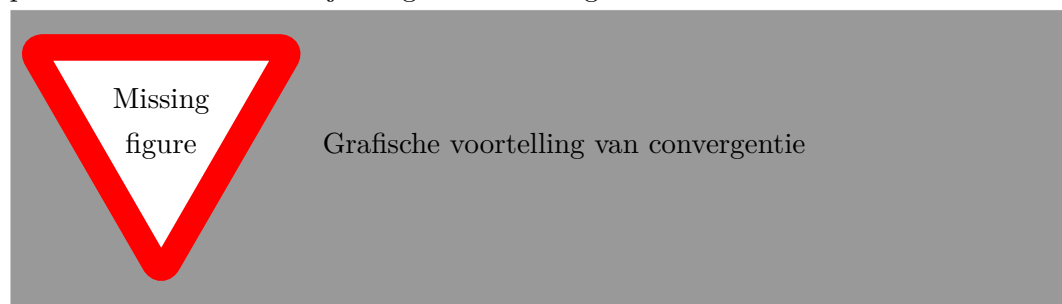
Bij het uitrollen van een configuratie (een "deployment run") inspecteert de tool de huidige toestand van elke machine en vergelijkt ze met de gewenste toestand. Als er een verschil is maakt de tool de nodige aanpassingen, indien niet onderneemt de tool geen actie. De beheerder van de verzameling systemen moet dus na het opstellen van de initiële configuratie zelf geen stappen meer ondernemen om te verzekeren dat de gewenste situatie bereikt wordt. Als er later nog aanpassingen moeten gebeuren moet enkel het model aangepast worden en een nieuwe deployment run gestart worden, manueel inloggen op elke server is niet meer nodig.

Een belangrijk aspect van elk gedistribueerd systeem zijn de afhankelijkheden die bestaan tussen de verschillende delen van dat systeem. In het voorbeeld van de LAMP-stack heeft de webserver naast PHP-mogelijkheden ook een werkende database nodig, anders kan deze niet alle functionaliteit aanbieden.



Als deze afhankelijkheden niet gespecificeerd worden in het model kan het configuratietool er ook geen rekening mee houden. De tool kan dus het model in een foute volgorde verwerken: eerst de webserver, dan php en uiteindelijk de database. De webserver zal bij het opstarten proberen te verbinden met de database, maar deze is nog niet online. Een op het eerste zicht succesvolle deployment run kan dus leiden tot een configuratie die niet volledig werkt.

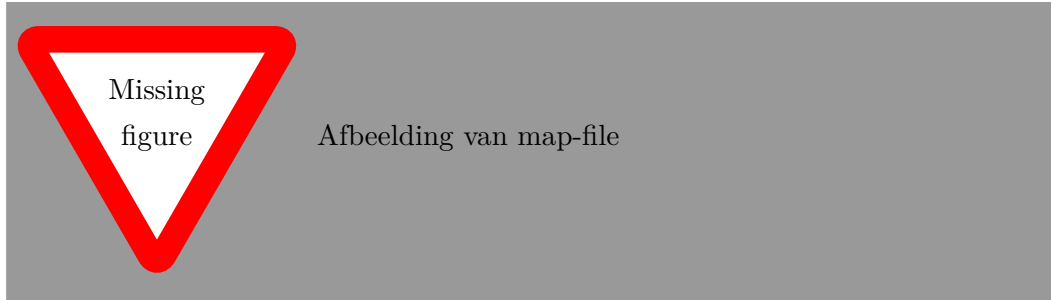
In vergelijking met de beginsituatie is de toestand van de setup na één run al minder afwijkend van de gewenste situatie; de tool zal nooit aanpassingen maken die zorgen voor een configuratie die verder afwijkt van het model dan voorheen. Na een paar iteraties zal dus altijd de gewenste configuratie bereikt worden.



Databases en webserver zijn abstracties die bestaan uit een verzameling objecten, een voorbeeld op een lager niveau is de afhankelijkheid tussen een bestand en de map waarin het staat: als de configuratietool eerst probeert het bestand te creëren en dan pas de map zal de deployment run slechts gedeeltelijk slagen want een bestand



kan niet bestaan zonder zijn parent folder.



We kunnen dus concluderen dat het vermelden van dependencies in het model het uitrolproces significant kan verbeteren, niet alleen in snelheid maar ook in correctheid.

De huidige configuratiebeheergereedschappen laten toe om op het niveau van bestanden, packages en services afhankelijkheden te specificeren. Bij het uitrollen van een model wordt dan een volgorde opgelegd waarmee de verschillende objecten verwerkt worden.

De tools die momenteel beschikbaar zijn compileren tijdens de deployment voor elke machine hun deel van het model. Elke machine krijgt dus informatie over wat hijzelf doet maar kan geen rekening houden met wat er op andere machines gebeurt. Afhankelijkheden binnen één machine verwerken is dus geen probleem maar afhankelijkheden tussen verschillende machines zijn niet mogelijk.

De tools die momenteel beschikbaar zijn kunnen dus het hierboven vermelde probleem van een webserver en een database niet oplossen. IMP (Infrastructure Management Platform) is een nieuwe tool die momenteel nog in ontwikkeling is. Een andere aanpak tijdens het deployen van een model laat toe afhankelijkheden tussen hoog-niveau objecten te specificeren: in tegenstelling tot de vorige tools krijgt elke machine het volledige model ter beschikking en niet alleen zijn eigen deel.

De doelstelling van deze thesis is een efficiënte manier vinden om van die extra informatie gebruik te maken en afhankelijkheden tussen hoog-niveau objecten om te zetten naar afhankelijkheden tussen de laag-niveau resources die zullen gedeployed worden.

Hier al eventueel consistency vermelden/verwijzen?

snelheid nog niet vermeld. Meerdere deployments onnodig maken.

Belang: correct gebeuren is van groot belang

Vermelden van workarounds



## Hoofdstuk 2

# Analyse van het probleem

In dit deel wordt er dieper ingegaan op de probleemstelling. Voor elke soort afhankelijkheid komt er een korte uitleg en daarna één of meerdere heuristieken die toepasbaar zijn.

### 2.1 Soorten afhankelijkheden

#### 2.1.1 Afhankelijkheden tussen bestanden en mappen

Hier gaat het over de afhankelijkheid tussen een bestand en de map waarin het staat. Een bestand kan niet bestaan zonder een map dus een heuristiek die elk bestand afhankelijk maakt van zijn parent folder is toepasbaar.

#### 2.1.2 Afhankelijkheden bij service stacks

Service stacks is een eigen term die ik gegeven heb aan een verzameling van een package en een service (en mogelijks ook een configuratiebestand). De verzameling wordt bepaald door te zoeken naar resources die binnen dezelfde scope gedefiniëerd zijn.

#### 2.1.3 Afhankelijkheden door relaties

IMP laat toe in het model relaties tussen hoog-niveau objecten te specificeren. Sommige relaties, zoals "x [1:] – [0:] y" impliceren dat y niet kan bestaan zonder een instantie van x. Zo bekomen we een heuristiek die de afhankelijkheid tussen x en y toevoegt aan het model.

#### 2.1.4 Afhankelijkheden tussen machines

In dit hoofdstuk wordt verder ingegaan op hoe afhankelijkheden tussen verschillende machines verwerkt worden. Het belangrijkste punt is hoe hoog-niveau relaties en afhankelijkheden vertaalt worden naar dependencies op de lagere niveaus.

## 2.2 Besluit van dit hoofdstuk

## Hoofdstuk 3

# Evaluatie

### 3.1 Besluit van dit hoofdstuk



## Hoofdstuk 4

### Besluit





# Bijlagen



# Bijlage A

## De eerste bijlage

In de bijlagen vindt men de data terug die nuttig kunnen zijn voor de lezer, maar die niet essentieel zijn om het betoog in de normale tekst te kunnen volgen. Voorbeelden hiervan zijn bronbestanden, configuratie-informatie, langdradige wiskundige afleidingen, enz.

In een bijlage kunnen natuurlijk ook verdere onderverdelingen voorkomen, evenals figuren en referenties[?].

### A.1 Meer lorem

#### A.1.1 Lorem 15–17

#### A.1.2 Lorem 18–19

### A.2 Lorem 51



## Fiche masterproef

*Student:* Harm De Weirdt

*Titel:* Configuratieafhankelijkheden gebruiken om gedistribueerde applicaties efficiënt te beheren in een hybride cloud

*Engelse titel:* Configuratieafhankelijkheden gebruiken om gedistribueerde applicaties efficiënt te beheren in een hybride cloud

*UDC:* T134

*Korte inhoud:*

### Context

Om IT infrastructures efficiënt te beheren wordt er gebruik gemaakt van configuratiebeheergereedschappen. Deze gereedschappen zijn model gebaseerd, waarbij het model de gewenste toestand van de configuratie beschrijft. Om de configuratie door te voeren wordt de gewenste toestand vergeleken met de huidige toestand en worden de nodige acties afgeleid die nodig zijn om de infrastructuur in die gewenste toestand te brengen. Huidige systemen zijn reeds in staat om eenvoudige afhankelijkheden af te leiden. Bijvoorbeeld dat een service eerst genstalleerd moet worden voordat die service gestart kan worden. Wat ontbreekt is afhankelijkheden tussen services op verschillende systemen in rekening brengen.

### Doel

Het doel van deze thesis is onderzoeken hoe afhankelijkheden in een configuratiemodel gebruikt kunnen worden om de initiele configuratie en mogelijke herconfiguraties van een hybrid cloud zo efficiënt mogelijk uit te voeren.

### Onderzoeksvragen

1. Hoe kunnen afhankelijkheden in een configuratiemodel gebruikt worden om veranderingen zo snel mogelijk uit te rollen?
2. Kan de gevraagde tijd gesimuleerd worden in functie van het configuratie model?

### Uitwerking

**Fase 1** Vertrouwd geraken met het configuratiebeheergereedschap

**Fase 2** Onderzoeken van bestaande configuratiemodellen

**Fase 3** Implementeren van een oplossing

**Fase 4** Valideren van de oplossing door middel van de configuratiemodellen op een private en publieke cloud en een simulator

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

*Assessoren:* Ir. W. Eetveel  
W. Eetrest

*Begeleider:* Ir. Bart Vanbrabant