

De Onderzoeksgroep
Software Languages Lab

nodigt U graag uit op de openbare verdediging van het proefschrift van

Mathijs Saey

ter behaling van de graad van Doctor in de wetenschappen

Titel van het proefschrift:

**Modular Distribution Strategies in
Stream Processing Frameworks:
a Meta-Programming Approach**

Promotor:

Prof. dr. Wolfgang De Meuter (VUB)

Co-promotor:

**Prof. dr. Quentin Stiévenart (Université
du Québec à Montréal, CA)**

De verdediging heeft plaats op

maandag 21 september 2026 om 17u

Campus Etterbeek VUB, Pleinlaan 2, Elsene in
auditorium I.O.01

Samenstelling van de jury

Prof. dr. Ann Nowé (VUB, voorzitter)

Prof. dr. Bas Ketsman (VUB)

Prof. dr. Pieter Libin (VUB)

Prof. dr. Annette Bieniusa (RPTU
Kaiserslautern-Landau, DE)

Dr. Pascal Weisenburger (Universität St.
Gallen, CH)

Curriculum vitae

Mathijs Saey behaalde zijn Master in de Computerwetenschappen aan de Vrije Universiteit Brussel, magna cum laude. Hij begon zijn doctoraat bij het Software Languages Lab onder begeleiding van Prof. Dr. Wolfgang De Meuter. Tijdens zijn doctoraat werkte hij mee aan de “Flamenco” en “Ecopipe” onderzoeksprojecten. Hij werkt momenteel mee aan het “Basecamp Zero” onderzoeksproject. Hij is coauteur van drie internationale journal publicaties (één als eerste auteur), twee internationale workshop publicaties (beide als eerste auteur), en stelde zijn werk voor op internationale conferenties en workshops. Hij begeleidde 5 Bachelor en 7 Master thesissen.

Abstract van het doctoraatsonderzoek

Onze wereld is overladen met toestellen die verbonden zijn met het internet en continu “events” produceren die door organisaties ontvangen worden als *stromen* van data. Veel organisaties willen zo snel mogelijk op deze data reageren. *Distributed Stream Processing Frameworks* (DSPFs) zijn software frameworks die het programmeren van applicaties die enorme volumes aan data verwerken vergemakkelijken. Deze frameworks staan deze applicaties toe om horizontaal te schalen door ze over verschillende machines in een “cluster-configuratie” te distribueren.

Essentieel voor de performantie van deze applicaties is de *distributiestrategie* die gebruikt wordt om de data over de nodes van de cluster te verdelen. In sommige DSPFs, zoals Apache Spark of Flink, zijn deze distributiestrategieën ingebakken in het framework, wat tot inefficiënte applicaties kan leiden. Op het andere eind van het spectrum vind je Apache Storm, dat een “low-level” model aanbiedt waarin programmeurs distributiestrategieën kunnen implementeren per applicatie om zo de applicatie-efficiëntie te verbeteren. Dit model verstrengelt echter de distributie en data logica van een applicatie, wat het moeilijk maakt om beide types logica aan te passen. Daarom moeten cluster programmeurs vandaag ofwel de ingebakken distributiestrategieën van een high-level framework accepteren, of omgaan met de complexiteit van deze strategieën uit te drukken in het low-level model van Storm.

Deze dissertatie introduceert Skitter: een nieuw programmeermodel dat metaprogrammeertechnieken gebruikt om data -en distributielogica in een stroomverwerkingsapplicatie van elkaar te ontwarren, en om het mogelijk te maken om nieuwe distributiestrategieën op een modulaire manier te implementeren. Dit maakt het mogelijk om stroomverwerkingsapplicaties in een high-level framework uit te drukken zonder de flexibiliteit van het low-level model van Storm te verliezen.

We stellen een formele beschrijving van Skitter, featherweightSkitter, en een praktische verwezenlijking van Skitter als een DSPF, Skitter.ex, voor. FeatherweightSkitter wordt samen met featherweightStorm, een nieuwe formele beschrijving van Storm, gebruikt om te bewijzen dat elke applicatie die in Storm uitgedrukt kan worden vertaald kan worden naar een equivalente applicatie in featherweightSkitter, wat aantoont dat de nieuwe abstracties “Storm-complete” zijn. Skitter.ex wordt gebruikt om aan te tonen dat Skitter de implementatie van distributiestrategieën uit de literatuur mogelijk maakt op modulaire wijze. Onze performantie-evaluatie van deze strategieën toont aan dat deze de verwachte performantiekarakteristieken vertonen en dat programma’s uitgedrukt in Skitter.ex “throughput” ratio’s vertonen in dezelfde grootorde als applicaties uitgedrukt in Storm.