



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Daniel Kirchner

Skalierbare Datenanalyse mit Apache Spark
Beispielimplementation eines Influenza-Frühwarnsystems

*Fakultät Technik und Informatik
Studiendepartment Informatik*

*Faculty of Engineering and Computer Science
Department of Computer Science*

Daniel Kirchner

Skalierbare Datenanalyse mit Apache Spark
Beispielimplementation eines Influenza-Frühwarnsystems

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Bachelor of Science Angewandte Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Kahlbrandt
Zweitgutachter: Prof. Dr. Zweitprüfer

Eingereicht am: 1. Januar 2345

Daniel Kirchner

Thema der Arbeit

Skalierbare Datenanalyse mit Apache Spark Beispielimplementation eines Influenza-Frühwarnsystems

Stichworte

Schlüsselwort 1, Schlüsselwort 2

Kurzzusammenfassung

Dieses Dokument ...

Daniel Kirchner

Title of the paper

Scalable Data Analysis with Apache Spark

Keywords

keyword 1, keyword 2

Abstract

This document ...

Inhaltsverzeichnis

1	Einführung	1
1.1	Motivation	1
1.2	Kontextabgrenzung	1
1.3	Relevante Produkte und Meilensteine	1
1.3.1	Überblick	1
1.3.2	Big Table	1
1.3.3	Map/Reduce	1
1.3.4	Hadoop	1
2	Vorstellung von Apache Spark	2
2.1	Übersicht	2
2.1.1	Architekturübersicht	2
2.1.2	Standardbibliotheken	2
2.2	Wesentliche Konzepte	2
2.2.1	Abgrenzung zu Hadoop	2
2.2.2	Resilient Distributed Datasets	2
3	Vorstellung des Beispiels	3
3.1	Aufgabenbeschreibung	3
3.2	Lösungsidee	3
3.2.1	1. Schritt: Ähnlichkeitsmaß für Wörter erzeugen	3
3.2.2	2. Schritt: Echtzeitbewertung von Textnachrichten aus einem Datenstrom	3
4	Implementation und Bewertung	4
4.1	Technischer Rahmen	4
4.1.1	OpenStack	4
4.2	Architekturübersicht	4
4.3	Architekturdetails	4
4.3.1	Modell für Ähnlichkeit von Wörtern mit MLlib erzeugen	4
4.3.2	Einlesen von Nachrichten aus dem Twitter Livestream	4
4.3.3	Verarbeiten und Bewerten der Nachrichten	4
4.4	Bewertung der Verfahren	4
5	Schlussbetrachtung	5
5.1	Kritische Würdigung der Ergebnisse	5
5.2	Ausblick und offene Punkte	5

Listings

1 Einführung

1.1 Motivation

1.2 Kontextabgrenzung

1.3 Relevante Produkte und Meilensteine

1.3.1 Überblick

1.3.2 Big Table

1.3.3 Map/Reduce

1.3.4 Hadoop

2 Vorstellung von Apache Spark

2.1 Übersicht

2.1.1 Architekturübersicht

2.1.2 Standardbibliotheken

Spark SQL

MLlib

Streaming

GraphX

2.2 Wesentliche Konzepte

2.2.1 Abgrenzung zu Hadoop

2.2.2 Resilient Distributed Datasets

3 Vorstellung des Beispiels

3.1 Aufgabenbeschreibung

3.2 Lösungsidee

3.2.1 1. Schritt: Ähnlichkeitsmaß für Wörter erzeugen

3.2.2 2. Schritt: Echtzeitbewertung von Textnachrichten aus einem Datenstrom

4 Implementation und Bewertung

4.1 Technischer Rahmen

4.1.1 OpenStack

4.2 Architekturübersicht

4.3 Architekturdetails

4.3.1 Modell für Ähnlichkeit von Wörtern mit MLlib erzeugen

4.3.2 Einlesen von Nachrichten aus dem Twitter Livestream

4.3.3 Verarbeiten und Bewerten der Nachrichten

4.4 Bewertung der Verfahren

5 Schlussbetrachtung

5.1 Kritische Würdigung der Ergebnisse

5.2 Ausblick und offene Punkte

See also [One und Two](#) (2010).

Literaturverzeichnis

[One und Two 2010] ONE, Author ; TWO, Author: A Sample Publication. (2010)

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 1. Januar 2345 Daniel Kirchner
