Master Thesis
Computer Science

UIMA, Docker and Kafka

Buzzwords oder doch interessant?

Simon Gehring

Am Jesuitenhof 3
53117 Bonn
simon.gehring@fkie.fraunhofer.de
Matriculation Number 2553262

At the
Rheinische Friedrich-Wilhelms-Universität Bonn
in cooperation with the
Fraunhofer-Institut für Kommunikation,
Informationsverarbeitung und Ergonomie

supervised by
Prof. Dr. Heiko Röglin and Daniel Töws

February 26, 2018

Contents

Chapter 1

Introduction

Natural language is part of everyone's everyday life and is most commonly used to transmit information human-to-human. While most of this interaction takes place orally or written on paper, the digital revolution and the rise of social media increased the amount of digitally stored natural language tremendously. Gantz and Reinsel predicted 2012 that the amount of digital data stored globally will double about every two years until at least the year 2020 [GR12].

Many opportunities arise from this amount of digital data, specifically in the field of machine learning. In 2011, IBM's QA (Question Answering) system "Watson" famously outmatched professional players in the quiz show "Jeopardy!" [Fer12, ESI+12]. Kudesia et al. proposed 2012 an algorithm to detect so called CAUTIs (Catheter-associated Urinary Tract Infections), common hospital-acquired infections, by utilizing a NLP (Natural Language Processing) analysis on the medical records of patients [KSDG12].

Apache UIMA (Unstructured Information Management Applications) is one of few general approaches to implement NLP solutions. With a very modular architecture, UIMA is a popular tool that can easily be applied to a majority of NLP problems. A large part of the popularity of UIMA stems from the large DKPro (Darmstadt Knowledge Processing Software Repository) collection of components, containing hundreds of analysis modules and precomputed language models [EdCG14], which are easily imported into existing Java projects by the build automation tool Apache Maven [Tea].

A common problem with UIMA in non-academic environments is scaling [DCR+15, ESI+12, RBJB+10]. UIMA itself provides two distinct interfaces to analyse larger collections of unstructured data, with one being UIMA-AS (UIMA Asynchronous Scaleout) and the other being the more dated and less flexible CPE (Collection Processing Engine) [FLVN09].

In this thesis, we will evaluate different means of scaling UIMA, using modern technologies like Docker, a container virtualization solution, Apache Spark, a cluster computing framework, and Apache Kafka, an information stream processing software. We will compare said implementations with the native UIMA-AS and CPE approach in terms of processor and memory efficiency, ease of implementation and maintainability.

1.1 Motivation

1.2 Basics

1.2.1 UIMA

1.2.2 Docker

1.2.3 Hadoop

1.2.4 Spark

1.2.5 Kafka

1.3 Problem

1.3.1 Scaling UIMA

UIMA-AS

UIMA-CPM

1.3.2 Implementation Requirements

1.4 Related Work

1.4.1 Watson

1.4.2 Something else that warrants another subsection

1.4.3 GATE?

1.5 Outline

Chapter 2

Implementation

2.1 Concrete Application

2.2 Documents -> Kafka

2.3 Kafka -> Spark

2.4 Spark -> UIMA

2.5 UIMA -> Java

2.6 Kafka -> Output

2.7 Bottlenecks

Chapter 3

Evaluation

3.1 Computation Speed

3.2 Memory Usage

3.3 Extensibility

3.4 Maintainability

Chapter 4

Summary

4.1 The Judgement

Chapter 5

Future Work

Glossary

Apache Kafka

Apache Kafka is an open-source stream processing software platform developed by the Apache Software Foundation written in Scala and Java. The project aims to provide a unified, high-throughput, low-latency platform for handling real-time data feeds. 3

Apache Spark

Apache Spark is an open-source cluster-computing framework. Originally developed at the University of California, Berkeley's AMPLab, the Spark codebase was later donated to the Apache Software Foundation, which has maintained it since. Spark provides an interface for programming entire clusters with implicit data parallelism and fault tolerance. 3

Catheter-associated Urinary Tract Infection

A urinary tract infection (UTI) is an infection involving any part of the urinary system, including urethra, bladder, ureters, and kidney. UTIs are the most common type of healthcare-associated infection reported to the National Healthcare Safety Network (NHSN). Among UTIs acquired in the hospital, approximately 75% are associated with a urinary catheter, which is a tube inserted into the bladder through the urethra to drain urine. Between 15-25% of hospitalized patients receive urinary catheters during their hospital stay. The most important risk factor for developing a catheter-associated UTI (CAUTI) is prolonged use of the urinary catheter. Therefore, catheters should only be used for appropriate indications and should be removed as soon as they are no longer needed. 3

Collection Processing Engine

UIMAprovides additional support for applying analysis engines to collections of unstructured data with its Collection Processing Architecture. The Collection Processing Architecture defines additional components for reading raw data formats from data collections, preparing the data for processing by Analysis Engines, executing the analysis, extracting analysis results, and deploying the overall flow in a variety of local and distributed configurations. 3

Darmstadt Knowledge Processing Software Repository

A collection of software components for natural language processing (NLP) based on the Apache UIMA framework. 3

Docker

Docker is a computer program that performs operating-system-level virtualization also known as containerization. It is developed by Docker, Inc. Docker is primarily developed for Linux, where it uses the resource isolation features of the Linux kernel such as cgroups and kernel namespaces, and a union-capable file system such as OverlayFS and others to allow independent "containers" to run within a single Linux instance, avoiding the overhead of starting and maintaining virtual machines (VMs). 3

Natural Language Processing

Natural-language processing (NLP) is a field of computer science, artificial intelligence concerned with the interactions between computers and human (natural) languages, and, in particular, concerned with programming computers to fruitfully process large natural language data. Challenges in natural-language processing frequently involve speech recognition, natural-language understanding, and natural-language generation. 3

Question Answering

Question answering (QA) is a computer science discipline within the fields of information retrieval and NLP, which is concerned with building systems that automatically answer questions posed by humans in a natural language. 3

UIMA Asynchronous Scaleout

UIMA-AS is the next generation scalability replacement for the Collection Processing Manager (CPM). UIMA-AS provides more flexible and powerful scaleout capability, and extends support to the UIMA-AS components not supported by the CPM, the flow controller and CAS multiplier. 3

Unstructured Information Management Applications

UIMA are software systems that analyze large volumes of unstructured information in order to discover knowledge that is relevant to an end user. An example UIM application might ingest plain text and identify entities, such as persons, places, organizations; or relations, such as works-for or located-at. 3

Bibliography

[DCR^{+}15] Guy Divita, M Carter, A Redd, Q Zeng, K Gupta, B Trautner, M Samore, and A Gundlapalli. Scaling-up nlp pipelines to process large corpora of clinical notes. *Methods of information in medicine*, 54(06):548–552, 2015.

[EdCG14] Richard Eckart de Castilho and Iryna Gurevych. A broad-coverage collection of portable nlp components for building shareable analysis pipelines. In *Proceedings of the Workshop on Open Infrastructures and Analysis Frameworks for HLT*, pages 1–11, Dublin, Ireland, August 2014. Association for Computational Linguistics and Dublin City University.

[ESI^{+}12] Edward A Epstein, Marshall I Schor, BS Iyer, Adam Lally, Eric W Brown, and Jaroslaw Cwiklik. Making watson fast. *IBM Journal of Research and Development*, 56(3.4):15–1, 2012.

[Fer12] David A Ferrucci. Introduction to "this is watson". *IBM Journal of Research and Development*, 56(3.4):1–1, 2012.

[FLVN09] David Ferrucci, Adam Lally, Karin Verspoor, and Eric Nyberg. Unstructured information management architecture (UIMA) version 1.0. OASIS Standard, mar 2009.

[GR12] John Gantz and David Reinsel. The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east. *IDC iView: IDC Analyze the future*, 2007(2012):1–16, 2012.

[KSDG12] Valmeek Kudesia, Judith Strymish, Leonard D'Avolio, and Kalpana Gupta. Natural language processing to identify foley catheter–days. *Infection control and hospital epidemiology*, 33(12):1270–1272, 2012.

[RBJB^{+}10] Cartic Ramakrishnan, William A Baumgartner Jr, Judith A Blake, Gully APC Burns, K Bretonnel Cohen, Harold Drabkin, Janan Eppig, Eduard Hovy, Chun-Nan Hsu, Lawrence E Hunter, et al. Building the scientific knowledge mine (sciknowmine):

a community-driven framework for text mining tools in direct service to biocuration. *Language Resources and Evaluation*, page 33, 2010.

[Tea] The DKPro Core Team. Dkpro core^{TM} user guide. https://zoidberg.ukp.informatik.tu-darmstadt.de/jenkins/job/DKProCoreDocumentation(GitHub)/de.tudarmstadt.ukp.dkpro.core$de.tudarmstadt.ukp.dkpro.core.doc-asl/doclinks/6/user-guide.html. Accessed: 2018-02-26.

Eidesstattliche Erklärung

Hiermit versichere ich, Simon Gehring, dass ich die vorliegende Masterarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Die Stellen meiner Arbeit, die dem Wortlaut oder dem Sinne nach anderen Werken und Quellen, einschließlich Quellen aus dem Internet, entnommen sind, habe ich in jedem Fall unter Angabe der Quelle deutlich als Entlehnung kenntlich gemacht. Dasselbe gilt sinngemäß für Tabellen, Karten und Abbildungen.

Unterschrift: ______________________________

Simon Gehring, Student
Universität Bonn