Process Discovery using Big Data stack - Implementing the Alpha Algorithm with Map-Reduce

Design Specification Document

Martin Hashem, Xiangan Chen

May 12, 2019
RWTH Aachen

1 Introduction

Process mining is an approach to extract process models from event logs. Since the distributed nature of modern information systems, event logs are likely to be distributed among different physical machines. Map-Reduce is a scalable approach for efficient computations on distributed data. In this Python application we will present the main idea of a Map-Reduce implementation of the Alpha process mining algorithm, to take advantage of the scalability of the Map-Reduce approach.

To fulfill the shortcoming in the current process mining technology, this project aims to build a new python-based web service, that integrates the big data capabilities of the Hadoop system into the process mining framework pm4py.

This document is presented for our shareholders of the project.
Within our project we mostly apply well-established open source applications, which includes the following tools with version numbers:

- Python 3.6 as back end programming language
- pm4py 1.1.10 as process mining toolkit
- Flask 1.0.2 as web framework
- Hadoop 3.2.0 as big data processing and distributed computation framework
- Docker CE as deploy platform

This very document is one of these, which will be provided to bring insights to our software development cycle:

- Project initiation document
- Requirements analysis document
- Design specification document
- Software documentation

2 System Overview

The main goal of this project is to create a web application that provides calculations with the Alpha algorithm and can prepare the data using the Map-Reduce.

The web service is to be accessed by a standard browser and needs to provide the following functionalities:

- Upload event logs in either CSV or XES format into the Hadoop system
- Front end links to existing algorithms for further processing
- Do Map-Reduce computations to run the alpha algorithm
- Download the files onto local system

3 Design Considerations

3.1 Assumptions and Dependencies

(Martin)

3.2 General Constraints

(Martin)

3.3 Goals and Guidelines

- Simple: One of the main guidelines through our system design is simple. From the frontend part to the backend, from the UIs to the layout, we set the simpleness on the crucial level.

- Resource-saving: The other goal of our design is resource-saving. Not only in our implement but also the outcoming software, we consume resources as little as possible, inclusive CPU computing time , storage and server occupancy.

3.4 Development Methods

Scrum is our first choice of development method. Since we have a small team, so an agile and intensive development method should fit us better. So instead of the classic waterfall model development, our team prefer Scrum. Scrum is a lightweight, iterative and incremental framework for managing product development. And enables our team to self-organize by encouraging physical co-location or close online collaboration of all team members, as well as daily face-to-face communication among all team members.[Wik19]

4 Architectural Strategies

5 System Architecture

6 Policies and Tactics

7 Detailed System Design

References

[Wik19] Wikipedia contributors. Scrum (software development) — Wikipedia, the free encyclopedia, 2019. [Online; accessed 12-May-2019].