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

Requirements Specification

Xiangan Chen, Martin Hashem

April 29, 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.

1.1 Purpose

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. The current scope of mining in Big Data is slow and centralized. Using map-reduce in combination with the Alpha algorithm, calculations can be deffered to more, smaller instances and also reduce overhead in transferring the events. Depending on the amount of map-reduce stages that are implemented, there can be significantly less time used during the mining process, due to the parallized nature of the approach. [GA14]

1.2 Scope

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 project contains these tasks:

- **Code required:** Intergrating the system Hadoop into the pm4py interface and read files from it

- **Code required:** Run the Alpha algorithm utilizing Map-Reduce
- **Code required:** Reading in logs and upload them into Hadoop

The web service is to be accessed by a standard browser and needs to provide the folowing 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

2 Specific Requirements

2.1 External Interface Requirements

User Interfaces Since this will be a web application, we will establish the interface on a python-based frontend framework called Flask. And the user interface should be clear and straightforward, given it can also be used by non IT professionals.

Hardware Interfaces There will be no strict constrains on Hardware because of the characteristics of convenience and cross-platform of a web application.

2.2 Functional Requirements

Alpha algorithm A paper by Assadipour[GA14] suggests that using a decentralized approach for process mining using the Alpha algorithm by splitting the calculations into smaller tasks available to smaller instances.
The Alpha algorithm takes eventlogs and generates a Petri net model including an initial and final mark. The main purpose of the Alpha algorithm is to portray the relations between activities, but comes with the flaw of not finding self loops.

Map-Reduce Map-Reduce is a technique to allow the user to have a scalable calculation on a distributed system. By first mapping events to an identifier and then reducing the map to its relevant values, the amount of needed data shrinks. Here the reductions prepare the data for the Alpha algorithm, removing the overhead of finding the traces first.

pm4py The pm4py library is a joint effort of the Fraunhofer FIT process mining group[pmg] and the RWTH Process and Data Science group[Pg], which supports state-of-the-art process mining algorithms in python.

Hadoop Apache Hadoop[had] is a collection of open-source software utilities that facilitate using a network of many computers to solve problems involving massive amounts of data and computation. It provides a software framework for distributed storage and processing of big data using the Map-Reduce programming model. So we take its advantages in our project so that we can deal with the big data and distributed event log properly.

Flask Flask[Fla] is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. It is very suitable for our web application.

2.3 Non-Functional Requirements

Performance Because of the effectiveness of Python language, this very web application will be performed fast responsive on the frontend. As for backend, since the hadoop is a distributed system framework, it will also be effizient.

Reliability Python and Java is the main language within our project, and they are supported actively.

Portability Because the web application is overall useable, whether on windows, linux or MacOS or on a cellphone.

Maintainability Our project team use the Git as a Version Control tool, and it will be updated and maintained regularly.

2.4 Logical Database Requirements

A Logical Database will be required, because we will be receiving CSV and XES data from the User, and the server will process the algorithm to gather the information that the user demand.

References

[Fla] Flask. http://flask.pocoo.org/docs/1.0/.
[GA14] Joerg Evermann Ghazal Assadipour. Big data meets process mining: Implementing the alpha algorithm with map- reduce. *ResearchGate Conferance*, March 2014.
[had] APACHE hadoop. http://hadoop.apache.org/docs/stable/.
[Pg] RWTH Process and Data Science group. http://www.pads.rwth-aachen.de/.
[pmg] Fraunhofer FIT process mining group. https://www.fit.fraunhofer.de/en/fb/risk/process-mining.html.