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

## Project Initiation

Martin Hashem, Xiangan Chen

April 20, 2019
RWTH Aachen

## 1 Overview

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.

## 2 Buisness Case

We develop a business case to disclose the potential of this project. The study is divided into three parts. First we explain one of the current algorithm in process mining. Furthermore we will define the scope of the project and point out the key benefits of our optimization.

### 2.1 Initial Situation

The current situation in process mining eventlogs using the Alpha algorithm is very centralized. All eventlogs are pushed together into one computation unit, eating up a big amout of resources, both in used space and computation time. The optimization of this is the main task.

### 2.2 Scope

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

The project contains these tasks:

- **Code required:** Integrating 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 standart browser and needs to provide the folowing functionalities:

- Upload eventlogs in either CSV or XES format to 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

Additionally these documentations will be provided to bring insights to our software development cycle:

- Project initiation document
- Requirement analysis document
- Design document
- Software documentation

### 2.3 Key Benefits

The current scope of mining in Big Data is slow and centralized. Using map-reduce in combination with the Alpha algorithm, callculations 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]

## 3 Feasibility Study

### 3.1 Theoretical Point of View

A paper by Assadipour[GA14] suggests that using a decentralized approach for process mining using the Alpha algorithm by splitting the callculations 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 is a technique to allow the user to have a scalable callculation 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.

### 3.2 Technical Point of View

For our project we mostly rely on well-established open source technology. This includes primarily the following tools:

- Python 3.7 as back end programming language
- pm4py python library as process mining toolkit
- Flask as web framework
- Hadoop as big data processing and distributed computation framework
- Docker as virtual machine and deploy platform

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. It is completely open source and intended to be used in both academic and industry projects.

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.

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.

Docker[de] is a computer program that performs operating-system-level virtualization. Docker is used to run software packages called containers. Containers are isolated from each other and bundle their own application, tools, libraries and configuration files; they can communicate with each other through well-defined channels. We will deploy our application on a production-grade server with Docker.

### 3.3 Risks and Mitigations

For our risk analysis we distinguish between project management related risks and technical risks. In order to reduce potential issues we define a list of strategies and countermeasures as general risk management.

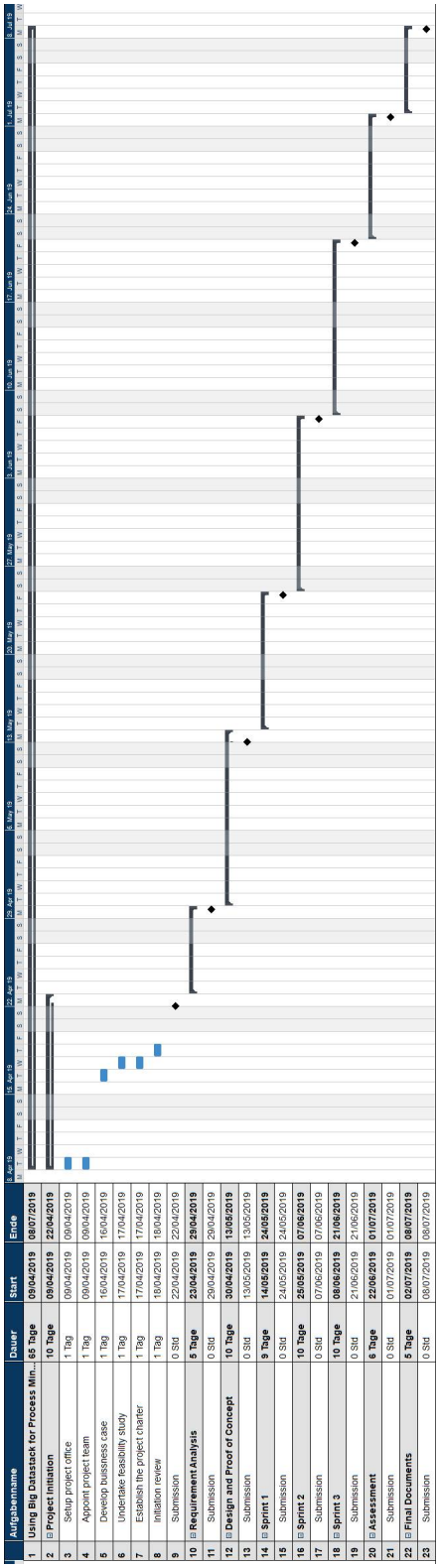
**Project Managment Risks** The following table covers potential organizational risks and countermeasures:

| Risk | Mitigation |
|---|---|
| Project team may misunderstand requirements. | Requirement misinterpretation can be mitigated by discussing between group members and frequent meetings with our advisors. Open questions should be addressed as soon as possible. |
| Learning curves lead to delays and cost overrun. | The degree of modularization depicted according to the project plan allows each team member to focus on a half set of skills only. |
| Time conflicts between team members. | Regular feedback communications. |
| Features are developed twice or not at all. | Since we have a two-man team, we will devide the tasks clearly and discuss them often. |

**Technical Risks** Since the Apache Hadoop and Flask are open source softwares, they bring us some significant advantages such as free to use, but they also have some limitations as compared to the commercial ones. In the following table technical related risks are considered:

| Risk | Mitigation |
|---|---|
| No formal support contracts. | Restrict to well-documented and established open source software. |
| Graphical user interface is not user-friendly. | Improve user interface in the frontend in an agile manner. |
| Software components do not work together. | Define clear and well structured interfaces in the early design phase. |
| Uploaded file not recognizable. | Appoint a obvious sign on the frontend site to notify the user to upload the correct format. |

# 4 Project Plan



| | Aufgabenname | Dauer | Start | Ende |
|---|---|---|---|---|
| 1 | **Using Big Datastack for Process Min...** | **65 Tage** | **09/04/2019** | **08/07/2019** |
| 2 | **Project Initiation** | **10 Tage** | **09/04/2019** | **22/04/2019** |
| 3 | Setup project office | 1 Tag | 09/04/2019 | 09/04/2019 |
| 4 | Appoint project team | 1 Tag | 09/04/2019 | 09/04/2019 |
| 5 | Develop buissness case | 1 Tag | 16/04/2019 | 16/04/2019 |
| 6 | Undertake feasibility study | 1 Tag | 17/04/2019 | 17/04/2019 |
| 7 | Establish the project charter | 1 Tag | 17/04/2019 | 17/04/2019 |
| 8 | Initiation review | 1 Tag | 18/04/2019 | 18/04/2019 |
| 9 | Submission | 0 Std | 22/04/2019 | 22/04/2019 |
| 10 | **Requirement Analysis** | **5 Tage** | **23/04/2019** | **29/04/2019** |
| 11 | Submission | 0 Std | 29/04/2019 | 29/04/2019 |
| 12 | **Design and Proof of Concept** | **10 Tage** | **30/04/2019** | **13/05/2019** |
| 13 | Submission | 0 Std | 13/05/2019 | 13/05/2019 |
| 14 | **Sprint 1** | **9 Tage** | **14/05/2019** | **24/05/2019** |
| 15 | Submission | 0 Std | 24/05/2019 | 24/05/2019 |
| 16 | **Sprint 2** | **10 Tage** | **25/05/2019** | **07/06/2019** |
| 17 | Submission | 0 Std | 07/06/2019 | 07/06/2019 |
| 18 | **Sprint 3** | **10 Tage** | **08/06/2019** | **21/06/2019** |
| 19 | Submission | 0 Std | 21/06/2019 | 21/06/2019 |
| 20 | **Assessment** | **6 Tage** | **22/06/2019** | **01/07/2019** |
| 21 | Submission | 0 Std | 01/07/2019 | 01/07/2019 |
| 22 | **Final Documents** | **5 Tage** | **02/07/2019** | **08/07/2019** |
| 23 | Submission | 0 Std | 08/07/2019 | 08/07/2019 |

## 5 Project Team

The project team consists of two persons. The following roles have been assigned to reflect individual strengths and competences:

| Name | Role | Contact |
|---|---|---|
| Martin Hashem | Project Manager | martin.hashem@rwth-aachen.de |
| | Backend Architect | |
| Xiangan Chen | Quality Manager | xiangan.chen@rwth-aachen.de |
| | Frontend Architect | |

In addition a team member is assigned to each milestone in the project, who is primarily responsible, that the requirements and goals of the milestone are met in time with sufficient quality. The responsibilities are stated in the following table:

| Milestone | Chief Responsibility |
|---|---|
| Project initiation document | Xiangan Chen |
| Requirement analysis document | Xiangan Chen |
| Design and proof of concept | Martin Hashem |
| Sprint 1 | Martin Hashem |
| Sprint 2 | Xiangan Chen |
| Sprint 3 | Martin Hashem |
| Assessment | Xiangan Chen |
| Final Documents | Martin Hashem |

## References

[de] docker enterprise. https://www.docker.com/.
[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.