

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

Martin Hashem, Xiangnan 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 Business 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

2.2 Scope

2.3 Key Benefits

3 Feasibility Study

3.1 Theoretical Point of View

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 3.6 as back end programming language
- pm4py python library as process mining toolkit
- Flask as web framewrok
- Hadoop as big data processing and distributed computation framework

The pm4py library is a joint effort of the Fraunhofer FIT process mining group and the RWTH Process and Data Science group, 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 is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries.[3] 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 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 MapReduce programming model. So we take its advantages in our project so that we can deal with the big data and distributed event log properly.

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 divide 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

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	
Xiangang 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	Xiangang Chen
Requirement analysis document	Xiangang Chen
Design and proof of concept	Martin Hashem
Sprint 1	Martin Hashem
Sprint 2	Xiangang Chen
Sprint 3	Martin Hashem
Assessment	Xiangang Chen
Final Documents	Martin Hashem

References

- [BDP93] Mic Bowman, Saumya K. Debray, and Larry L. Peterson. Reasoning about naming systems. *ACM Trans. Program. Lang. Syst.*, 15(5):795–825, November 1993.
- [Bra91] Johannes Braams. Babel, a multilingual style-option system for use with latex’s standard document styles. *TUGboat*, 12(2):291–301, June 1991.
- [Cla91] Malcolm Clark. Post congress tristesse. In *TeX90 Conference Proceedings*, pages 84–89. TeX Users Group, March 1991.
- [Her93] Maurice Herlihy. A methodology for implementing highly concurrent data objects. *ACM Trans. Program. Lang. Syst.*, 15(5):745–770, November 1993.