

MapReduce-System (37)

Fabian Kleinrad (07), 5BHIF

March 2022

Contents

1	Introduction	2
2	MapReduce	2
2.1	Map	3
3	Class-diagram	3
3.1	Classes	3
4	Implementation	3
5	Usage	3
5.1	Command Line Arguments	3
5.1.1	Configuration	3
6	Project Structure	4

1 Introduction

In this project the technology MapReduce is being simulated. Thereby a simple system has been developed to imitate a the functionality of an MapReduce application. All of the functionality in this project is written with C++17 and compiled with the help of the meson build system¹. The communication is based on the TCP protocol and realized using the C++ library asio².

Furthermore to increase performance and usability protocol buffers³ are utilized. Protocol Buffers enable the serialization of data structures in an efficient manner, which simplifies working with messages sent between parties in the MapReduce system.

To make use of the advantages of using a MapReduce architecture, a simple use-case consisting of counting the number of character occurrences in a plain text document. This kind of application was chosen due to its simplicity, which enables the focus of this project to stay on MapReduce rather than a test application.

2 MapReduce

MapReduce is a programming model developed to decrease computation time of large data sets. It was invented by Google, the reason being the need to compute various kinds of derived data. Examples would be inverted indices or representations of the graph structure of web documents. These applications all have simplicity in common, there are no complex operations needed to accomplish said tasks. Furthermore are these kinds of processes characterized by accepting large amounts of input data and reducing it to fraction of itself. MapReduce presents a solution to parallelization, fault-tolerance, data distribution and load balancing. Thereby it is based on the principle of map and reduce, which are eponymous for the technology.⁴

¹*The Meson Build system.* URL: <https://mesonbuild.com> (visited on 03/30/2022).

²*asio C++ Library.* URL: <https://think-async.com/Asio/> (visited on 03/30/2022).

³*Protocol Buffers.* URL: <https://developers.google.com/protocol-buffers> (visited on 03/30/2022).

⁴URL: <https://static.googleusercontent.com/media/research.google.com/de/archive/mapreduce-osdi04.pdf> (visited on 04/03/2022).

2.1 Map

The map function accepts a set of key/value pairs, with the implementation being provided by the user.

3 Class-diagram

3.1 Classes

4 Implementation

5 Usage

5.1 Command Line Arguments

5.1.1 Configuration

6 Project Structure

References

- [1] URL: <https://static.googleusercontent.com/media/research.google.com/de//archive/mapreduce-osdi04.pdf> (visited on 04/03/2022).
- [2] *asio C++ Library*. URL: <https://think-async.com/Asio/> (visited on 03/30/2022).
- [3] *Protocol Buffers*. URL: <https://developers.google.com/protocol-buffers> (visited on 03/30/2022).
- [4] *The Meson Build system*. URL: <https://mesonbuild.com> (visited on 03/30/2022).