INTRODUCTION TO COMPUTER SYSTEM DESIGN

Johan Sebastian Arias

August 2020

1 Introduction

This is a program developed in Java in conjunction with SparkWeb and Git which calculates the mean and standard deviation given a data set of n real numbers. The main objective of this project is based on identifying the proposed architecture for the implementation of an integrated web service with different technologies such as Apache Spark which is a framework that allows us to manufacture our web resources in impressive response times and with a simple complexity. Heroku is in charge of delivering a deployment of our web application, Maven distributes the dependencies in an organized way to build our project, GitHub is in charge of the version management and CircleCI has the responsibility of maintaining a continuous integration for any change in our project.

2 Objectives

1. To create a simple web application with Sparkweb and recognize its advantages.
2. To learn implementing complex systems.
3. To understand the cycle of a web application deployed.
4. To understand the basics of Maven,Git,Heroku,Circleci via command line.
5. To learn and implement continuous integration and realize of its importance in real environments.
6. To understand how a simple web server is built from scratch.
7. To identify the components of a client-server based architecture.

3 Definitions and Context

3.1 SparkWeb Overview:

3.1.1 What is SparkWeb?

Apache Spark is an open source cluster computing framework for real-time data processing. The main feature of Apache Spark is its in-memory cluster computing that increases the processing speed of an application. Spark provides an interface for programming entire clusters with implicit data parallelism and fault tolerance. It is designed to cover a wide range of workloads such as batch applications, iterative algorithms, interactive queries, and streaming.

Figure 1: Apache Spark features
[2]

3.1.2 Sparkweb Architecture Overview

Apache Spark has a well-defined and layered architecture where all the spark components and layers are loosely coupled and integrated with various extensions and libraries. Apache Spark Architecture is based on two main abstractions: [2]

1. Resilient Distributed Dataset (RDD).
2. Directed Acyclic Graph (DAG).

But what those really mean?, basically RDD's are collections of data items that are split into partitions and can be stored in-memory on workers nodes of the spark cluster. RDD stands for:

- **Resilient:** Fault tolerant and is capable of rebuilding data on failure.

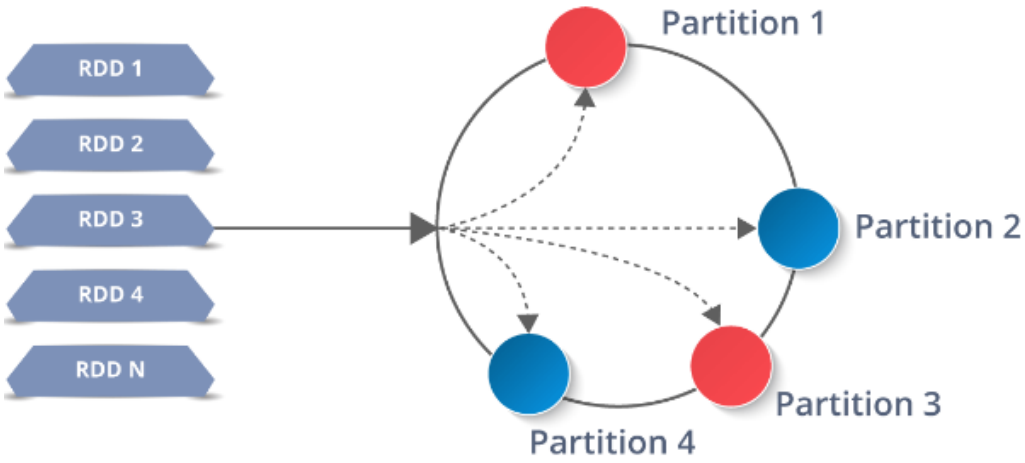

Figure 2: RDD Graphic representation
[1]

- **Distributed:** Distributed data among the multiple nodes in a cluster.
- **Dataset:** Collection of partitioned data with values.

Apache Spark follows a master/slave architecture with two main daemons and a cluster manager.

- Master Daemon – (Master/Driver Process).
- Worker Daemon –(Slave Process).

4 Design of this project

4.1 Design's description:

Dentro del diseño de este proyecto se pueden identificar diferentes elementos o abstracciones como compiladores que en el contexto de este programa a alto nivel son :

- **SparwebApp: (Server-side)** This component receives and responses with the original build of the webpage.
- **Java Virtual Machine JVM (Server-Side) :** After a request has been passed the typical cycle coming from the browser, the JVM uses the operating system modules that are brought to the processor as the end point.
- **Operating System OS (Server-side/Client-side):** This is the component in charge of providing an interface to the user applications.
- **Processor CPU (Server-side/Client-side) :** This is a physical component that performs the task of processing an input and delivering an output.
- **Browser (Client-Side):** It is the software that allows access to the Internet network, also allows us to generate requests.

As communication link abstractions we can identify the requests that are made through the **HTTP** protocol (**Request and Response**) which allow us to communicate the client with the server. Similarly, a communication link exists between the operating system, the applications and the processor.Since we are using the spark framework, we can therefore conclude that we are using an architecture defined by the client-server model.

4.2 Architecture Diagrams

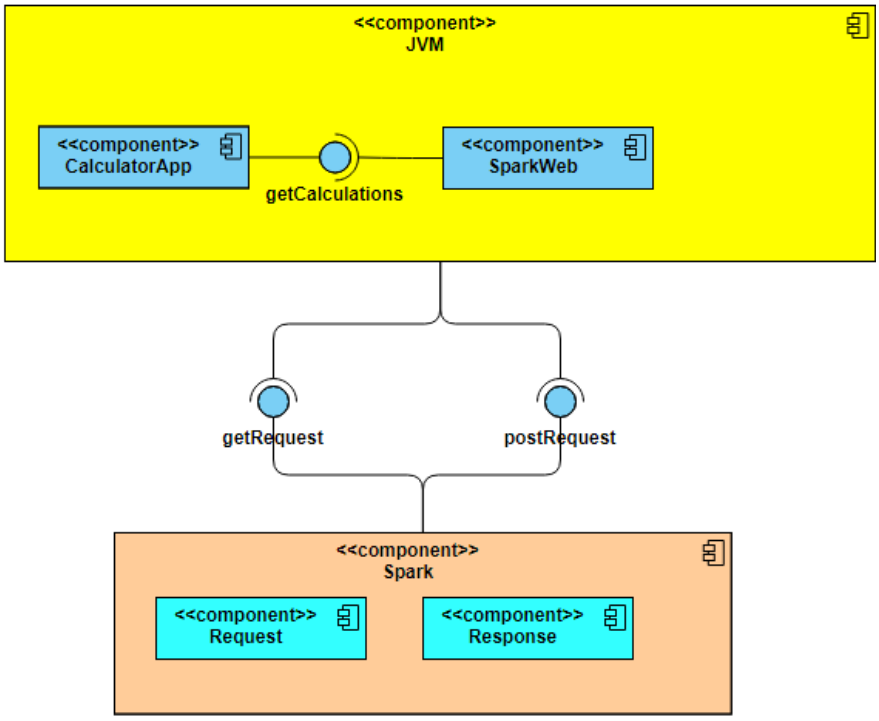

Figure 3: Components diagram of this program

4.3 Class diagram:

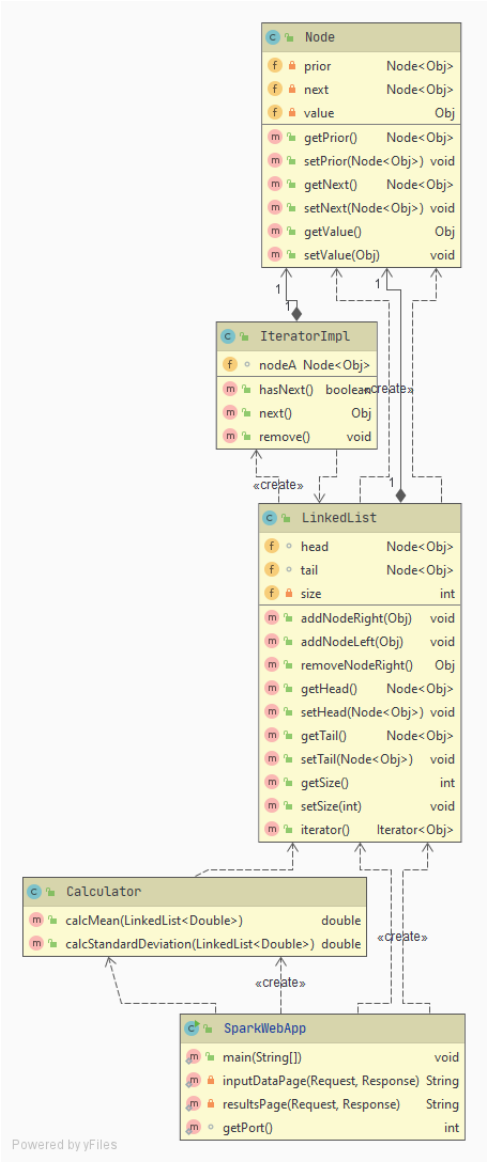

Figure 4: Class Diagram of the design

5 Setting up this project at your local computer

Requisites

- Apache Maven
- Java 8
- git

1. Clone this repository:

   ```
   git clone https://github.com/JohanS11/LAB2-AREP.git
   ```

2. Build the project with maven:

   ```
   cd LAB2-AREP && mvn package
   ```

3. Execute the project with maven:

   ```
   mvn exec:java -Dexec.mainClass="edu.eci.arep.sparkwebapp.SparkWebApp"
   ```

4. Now you should be able to run this project locally at **http://localhost:4567**

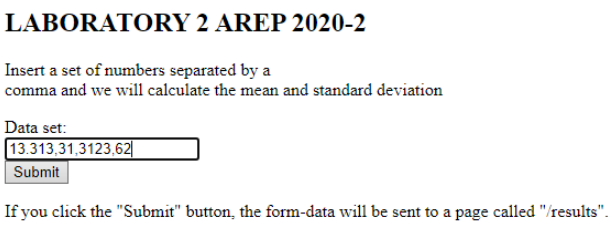

Figure 5: Running app locally

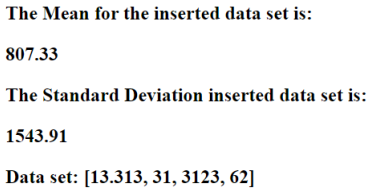

Figure 6: Results at the "/results" resource

6 Conclusion

From this project I have identified the architecture behind the deployment of a web server. We have also identified the components involved in building a web application. I have learned the use of the spark framework and the incredible advantages it brings.

References

[1] Apache spark architecture explained in detail. `https://www.dezyre.com/article/apache-spark-architecture-explained-in-detail/338`. Accessed on 2020-08-19.

[2] Apache spark architecture – spark cluster architecture explained. `https://www.edureka.co/blog/spark-architecture/`. Accessed on 2020-08-19.