Final Report of Wireless Communication and Mobile Network

A Display Website with Spark APIs

Chang Zhou
StudentID:5130309787
Class:F1303027
School of Electronic Information and Electrical Engineering
Shanghai Jiao Tong University
lemonbirdy@sjtu.edu.cn

2016/06/10

Abstract

In this semester, I continued on learning spark and started to learning play framework. Meanwhile I studied some theory on similarity calculation called SimRank. For presentation, I developed a website to show some Spark APIs that our group designed. In this report, I would mainly talked about my development of the website which included the application of both Play framework and Spark programming.
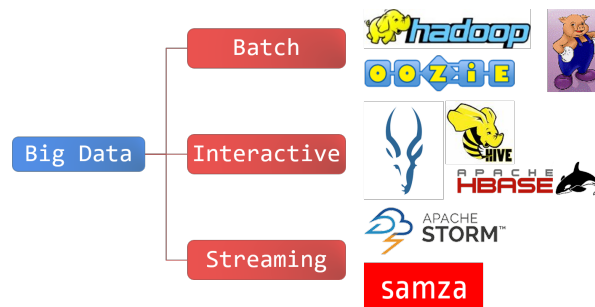
Chapter 1

Introduction

In this big data era, we have massive resource. How to unearth these buried treasure becomes one of the key issues today. Many data mining tools are coming thick and fast. Some deal the big data through batch, like Hadoop, pig. Some provide interactive distributed database like impala, hive. Some specialize the process of streaming data like storm, samza. A big-data developer has to master at least three or more tools in order to utilize the mountainous data. In this way, Spark came into being. Within a single framework, Spark integrated all these three parts. Also it has a good support for most of those popular big-data tools.

Spark is proud of its fast speed. Compared to Hadoop, it has a real advantage of 10 times faster over Hadoop on computation. Developing also becomes concise when using spark. One sentence completes the job of wordcount which others may use more than 80 rows. Spark also provides us with plenty of APIs, like MLlib, GraphX, which made it highly multifunctional.

As to Spark, such a powerful distributed computation framework, we can make good use of it to speed up our compuation of some algorithm. My work is mainly about translating some local programs into distributed programs through Spark. So the question how to display my work inspired me to develop a website.

Chapter 2

Motivation

Since I started to learn Spark, there have been always two questions confused me. As a new comer: why shall I choose Spark? From installation to my first app, I spent totally three days. It's quite difficult to start to use.

As a developer, how can I show my contribution? No complex theory, no pretty interface, all we have is millions of bugs. Is there anyone who likes a presentation that talks about how someone overcame one bug after another.

Therefore I came up with an idea to develop a website. So what can a website do? There are my three motivations.

2.1 Intuitive

New comers can find instructions and tutorial blogs on our website. They can experience spark's fast computation through some online examples without any installation and compilation. Besides the offical gudieline, here would be the best for Spark developers to study Spark.

2.2 Contributive

Developers can put their spark programs on the website. Anyone who visited our website can take a try on our APIs. It not only offers a display platform for us developers to show our work and effort, but also encourages us to contribute more work. The suggestion from user and the pressure to publish may push us to make development better and better.

2.3 Impressive

This is based on the former two. Our website is also desired to face other spark developers. They come to our website to get new ideas, to communicate with each other, and to obtain instructions. I want to make our website more

professional and multifunctional. It is not only a presenation platform for SJTU to display its work and contribution on Spark, but also a community for all the Sparker to make joint effort to utilize and improve Spark.
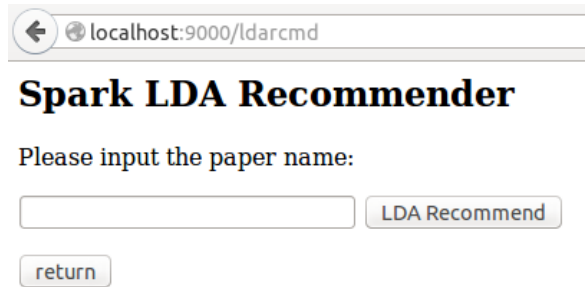
Chapter 3

Our Website

3.1   Interface
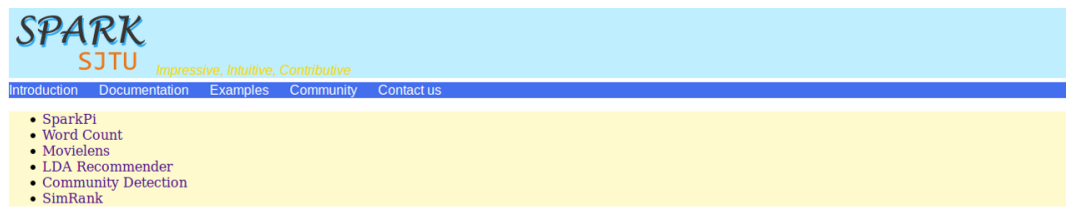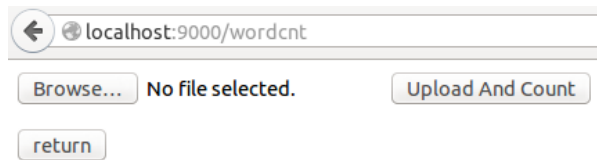
3.2 Function

3.2.1 Word Count

localhost:9000/ldarcmd

Spark LDA Recommender

Please input the paper name:

[   ] [LDA Recommend]

[return]

3.2.2 LDA Recommender

localhost:9000/wordcnt

[Browse...] No file selected.   [Upload And Count]

[return]

Chapter 4

Play Framework

I use play framework to build the website. Play is a high-productivity Java and Scala web application framework that integrates the components and APIs you need for modern web application development. Play is based on a lightweight, stateless, web-friendly architecture and features predictable and minimal resource consumption (CPU, memory, threads) for highly-scalable applications thanks to its reactive model, based on Akka Streams. Why I use Play Framework? There are two main reasons.

Play framework completely supports the Scala programming. The spark is written in Scala, it becomes possible to directly use the spark API as a function. No other interfaces is needed.

Play framework provides a characteristic development of html file, I can use scala to program or call a scala function just by adding a prefix symbol @. If I want to output a list in the webpage, I can just use scala's for-loop.

Then let's go a little deeper to see how play framework works:

To get a webpage, it will first make a call to the Routes table. It searches the table to find whether there is a url matches it and get the corresponding control object, for example, 'control.Application.index'. Then, it visits the control object. There may be some operations. After that, it returns the view object with Result type, for example, 'Ok', 'Redirect'. Finally the view object tells the browser to show something through html or scala.
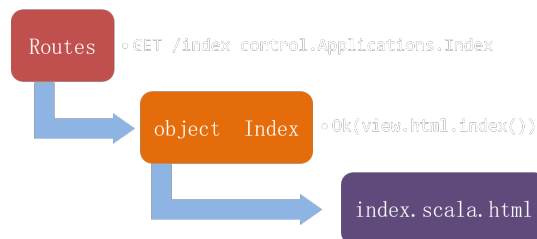
Chapter 5

Spark APIs

5.1 Word Count

The first example is word count. It reads a textfile, and then counts the times each word appears.

```scala
def wordcnt(filename: String) : Array[(String, Int)] = {
    val sc = new SparkContext("local", "WordCount")
    val line = sc.textFile("resources/tmp/"+filename)
    val cnt = line.flatMap(_.split(" ")).map((_, 1)).reduceByKey(_ +
        _).collect()
    sc.stop()
    return cnt
}
```

5.2 Simple Recommender on LDA

First use LDA to train a topic model. The spark's LDA function provided me with a table contains each paper's topic contribution.
Secondly I use Euclidean distance to calculate the similarity between the searched paper and others.
Thirdly, I get the similarity result and take the first n items to show.
Paper Topic Distribution Computation:

```scala
def getTopic(pathname: String, clusterNum: Int) : Array(Int,
    Array(Int))= {
    val sc = new SparkContext("local", "getTopic",
        System.getenv("SPARK_HOME"),
        Seq(System.getenv("SPARK_TEST_JAR")))
    val data = sc.textFile(pathname)
```

```
    val parsedData = data.map(s => Vectors.dense(s.trim.split('
        ').map(_.toDouble)))
    val corpus = parsedData.zipWithIndex.map(_.swap).cache()
    val ldaModel = new LDA().setK(clusterNum).run(corpus)
    val tTPD = ldaModel.topicDistributions.collect().sortBy(t => t._1)
    return tTPD
}
```
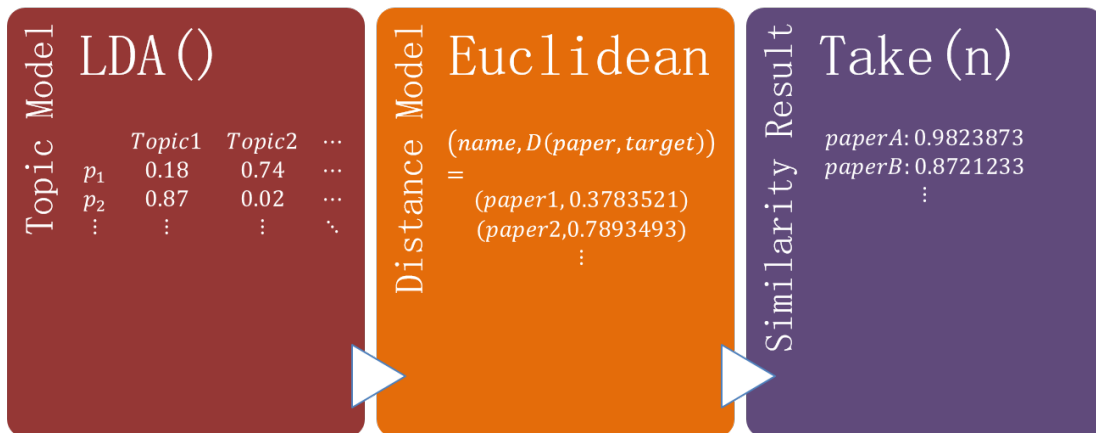
Euclidean Distance Calculation:

```
def distance(x: Vector, y: Vector) = {
    math.sqrt(x.toArray.zip(y.toArray).map(p => p._1 - p._2).map(d =>
        d*d).sum)
}
```

Chapter 6

Future Work

1. Prettify the Website
   As you see, our website still remains simple. Not only the APIs need a more suitable interface, the web page itself has also somewhere to improve.

2. Add Details
   Some details, like documents, hasn't been added yet. It will be a trivial work.

3. A More Reasonable Management Frame of APIs
   We can't make our APIs just in a list. A more reasonable structure is needed to manage these APIs.

Chapter 7

Thank

Chapter 8

Reference

http://spark.apache.org/
https://www.playframework.com/