CSSE 490: PROJECT CONNECTRIX

PROBLEM STATEMENT

ZHIHAO XUE, TIANJIAO MO, (JACKIE) XIANGQING ZHANG

TABLE OF CONTENTS

REVISION HISTORY

Date	Comment
2015/9/22	Initial Draft Proposal
2015/9/23	Document Structure
TBA	TBA

INTRODUCTION

People are interested in Internet. One of the things they cannot easily find is how their favorite websites are connected. Connectrix, the project we are proposing, aims to visualize the relationship between websites. Users can see relation graphs on the web page, as well as other statistics displayed. We will use the Common Crawl[1] dataset as our main data set, and we will run real-time crawlers to parse fresh data to update the knowledge base.

FEATURES

READING DATA FROM DATASET

We have chosen graph as our preferred datatype and that will also be how data is stored in our system, but we will also take other datatypes like JSON[2] and Tree. We would have different functions to parse different datatypes. Each piece of raw data in the given JSON dataset should be a key/value pair in which the key is a website URL and the value is a list of URLs of those websites that connected to the website denoted by the key. In a tree every connected websites to a certain website are children of the node which denotes that website.

READING DATA FROM WEB CRAWLER

We will kick off a really simple web page crawler that starts from a given URL web page, scans all valid external links in the page (by Regex-ing[3] the *a* tags), inspects other page details, and generates an raw input using an easy-to-parse data format, such as the one mentioned above. Continuing from all available URLs the web crawler has got so far, it will perform a BFS-like search to continue crawling data.

The web crawler will only send GET requests without any cookies and with the browser identification of standard Mozilla browser. If it cannot fetch a website (i.e. status code other than 200), it will simply ignore the current URL and move on.

PARSE DATA THROUGH HADOOP

We will use Falcon to perform data pipeline process.

In the data pipeline process, we will have two types of clusters. The first one is raw-data cluster where the information about website is parsed and the raw data (html in this case) is stored. Then the raw data will be processed to get data cleanse which can be converted to clean data (clean htmls in this case). The second type is clean-data cluster on which the clean data is stored.

DISPLAY RESULTS ON WEB PAGES

We will host a website to provide user with the access to these graphs. To be specific, user can open the website, and enter either one or two website domains.

1. If user enters only one website domain, we will display the first top 200 websites that are linked to and linked from the user input website. These websites are sorted in link depth order, which is the webpage's URL depth from root domain. If more than 200 websites are presented, the extras will not be displayed. Other statistics, such as the known number of websites that are linked to and linked from this website, will also be displayed.
2. If user enters two website domains, we will determine how they are connected. The page will display some pages where forward links to another site present, what is the earliest time they connect, average depth of links, and so on.

IMPLEMENTATION

LANGUAGE

We decide to use Java for Hadoop side coding. Two reasons behind are:

- **Java is the common coding language among all team members**. Starting from CSSE220, we have been exposed in Java environments for a long time. So using Java reduces the cost of learning new languages.
- **Java works natively with Hadoop**. Since Hadoop is written in Java, to avoid any uncertainties caused by using new languages, we should better stick with Java.

On the web application side, we decide to use Node.js + HTML + CSS + JavaScript. This is the standard practice that are widely used amongst web applications.

TOOLS

We decided to give Falcon a try. Apache Falcon is a framework to simplify data pipelining and management on Hadoop clusters. In general, it eases the pain of create new workflows and/or pipelines, supports large data handling process, and provides feedback from pipelines.

For more information, please refer to Research 1 Documentation[4].

STAKEHOLDERS

Stakeholder Role	Name
Professor & Instructor	Sriram Mohan
Developer	Zhihao Xue
Developer	Tianjiao Mo
Developer	(Jackie) Xiangqing Zhang
TBA	TBA

REFERENCES

[1] Common Crawl, http://commoncrawl.org/, obtained Sep 2015.

[2] JSON (JavaScript Object Notation), http://www.json.org/, obtained Sep 2015.

[3] Regex (Regular Expression), https://en.wikipedia.org/wiki/Regular_expression, obtained Sep 2015.

[4] Research 1 Documentation, *Project Connectrix*, Sep 2015.