Distributed Crawler and Big Data Applications

Li Wenbo
Project: A Distributed Machine Learning Platform
1155077143@link.cuhk.edu.hk

Introduction

Inplementation of High-Performance Distributed Web Crawler and Big Data Applications

I am belonging to crawler group; whose task aims at providing plenty of raw data to build a big data foundation so that other group members (e.g. machine learning group) can train their specific mode or test their existing algorithms.
Meanwhile, some attempts related with big data can be efficiently accomplished by utilizing crawled data.
With the help of high efficient system Husky, billions of data can be downloaded in few minutes.

Basic Crawler Implementations

Methods in Web Crawler

- First method makes use of URL to distinguish webpage differences by number changes. We can slightly change the number so that we can link to adjacent webpages.

```
def trade_price(max_pages):
    page=1
    while page<=max_pages:
        url='http://sz.lianjia.com/ershoufang/luohu/pg' + str(page)
+ '/'
     page += 1
```

- Second method firstly insert the main websites of crawling object, and define the limit requesting time for each page, then assign proxy servers and make arrangements for RE search rules in order to link to wanted resource webpages, finally download related data to HDFS files and repeat the whole process.

- Main functions are defined in hcrawl, and it is quite simple for user to import it in program and follow steps to crawl:

```
__init__.py
__init__.pyc
crawler_config.py
crawler_config.pyc
crawler_dist.py
crawler_dist.pyc
crawler_utils.py
crawler_utils.pyc
```

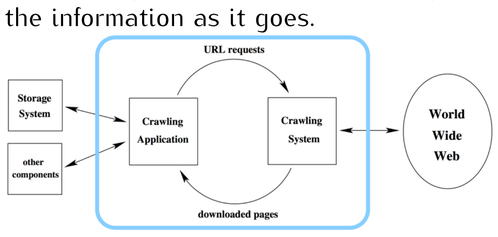
The Distributed Web Crawler Approach

Master-Worker Architecture

- A Husky cluster consists of one master and multiple worker. The master is responsible to coordinate the workers, and workers perform actual computations.

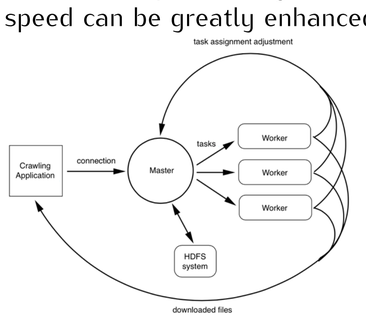
Crawler Operation Principles

- Generally, a web crawler starts with a list of URLs to visit. As the crawler visits these URLs, it identifies all the hyperlinks in the page and adds them to the list of URLs to visit, called the crawl frontier. URLs from the frontier are recursively visited according to a set of policies. If the crawler is performing archiving of websites it copies and saves the information as it goes.



Crawler Distribution with Husky

- With the help of Husky, the task of crawler can be distributed to workers so that the speed can be greatly enhanced.



Results Comparison

- In single local server, tested program can reach approximately 5 to 10 HTML pages per second. By comparison, a configuration as the one in Figure 3 would require between 3 to 5 workstations, and would achieve an estimated peek rate of 250 to 400 HTML pages per second.

City Super Customer Impression Analysis

Data Description

- By web crawler with Husky, customer comments from Da Zhong Dian Ping website can be easily downloaded. We specify one store of City Super located in IAPM, retrieving its comments from 2014.5.1 to 2016.5.25. Totally three data sets with 42,955 Chinese words. Then convert the comments into English, and design algorithm to realize word count and word selection (e.g. exclude common words is, the etc.).

Data Visualization

Observation and Forecast

- By carefully observing the data visualization results, it is obvious to figure out the main impression of customer toward IAPM store: Good, Fresh, Imported, Price, etc. And this just accords with City SuperâĂŹs core ideas. Also, by comparing words through different years, it is clear to find out the trend of customersâĂŹ impression (e.g. word âĂIJJapaneseâĂİ becomes smaller and smaller, it means the attention of customers towards Japanese product decreased). Store manager may adjust the product model to comply with trend.

Data Visualization for Conference Analysis

Data Description

- By powerful crawler with Husky, we can collect Interspeech (2014) conference papers. After arranging, Interspeech 2014 conference released 12 main tracks, 1,078 paper abstracts. The total number of paper abstracts is 1,078, words of which are 101,312. After pre-processing in terms of stemming and removal, the vocabulary contains 5,732 unique words. Each document contains 200 words with specific probabilities both in percentage form and exponential form varying from 2000 to 2015.

LDA

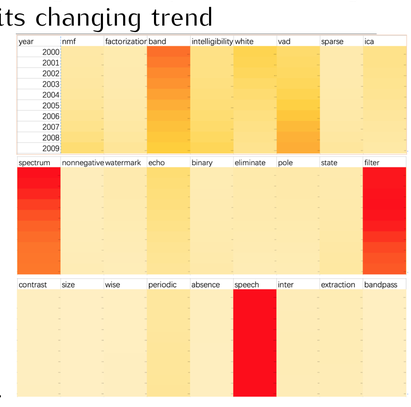
- LDA is a topic probability model producing distinct sets of data like text corpora.

$$p(D\,|\,\alpha,\beta)=\prod_{d=1}^{M}\int p(\theta_d\,|\,\alpha)\left(\prod_{n=1}^{N_d}\sum_{z_{dn}}p(z_{dn}\,|\,\theta_d)p(w_{dn}\,|\,z_{dn},\beta)\right)d\theta_d.$$

Data Visualization

- In Picture we can check the correlation degree between expert-defined tracks and latent topics.

- Using heat map, we can easily locate the high popularity words, also we can figure out its changing trend



.

Observations and Forecast

- By observations we can figure out that Speech keeps rising and has the highest value, so further interest should be paid; Signal has high value but keeps decreasing, so the concentration for next yearâĂŹs conference tracks regarding Signal may reduce. Each topic can be analyzed in this method and evidence-based predictions can be extracted from data ultimately.

References

- F. Yang, J. Li, and J. Cheng. Husky: Towards a more efficient and expressive distributed computing framework. PVLDB, 9(5):420–431, 2016.
- P. Liu, S. Jameel, W. Lam, B. Ma, and H. M. Meng. Topic modeling for conference analytics. In INTERSPEECH, pages 707–711, 2015.