Future Generation Computer Systems 59 (2016) 33–46

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Parallelizing the extraction of fresh information from online social networks

Rui Guo, Hongzhi Wang *, Mengwen Chen, Jianzhong Li, Hong Gao

Harbin Institute of Technology, China

HIGHLIGHTS

- Category OSN users into 4 types according to their post behavior.
- Proposal the Poisson process model and hash model to collect fresh tweets.
- Discuss the parallelization technology of the Poisson process model.
- Design the centralized and distributed architectures of the crawler system.
- Conduct extensive experiments to verify the models and architectures.

ARTICLE INFO

Article history:
Received 15 March 2014
Received in revised form
8 October 2015
Accepted 18 November 2015
Available online 9 December 2015

Keywords:
Crawler
Freshness
Online social network

ABSTRACT

Online social networks (OSNs) are among the hottest new services in recent years. OSNs maintain records of the lives of users, thereby providing potential resources for journalists, sociologists, and business analysts. Crawling data from social networks is a basic step during the processing and analysis of social network information. However, as OSNs become larger and the information on the network updates faster than the web pages, crawling is more difficult due to limitations in terms of bandwidth, politeness or etiquette, and computational power. To extract fresh information from OSNs in an efficient and effective manner, we propose a novel method for crawling and we also discuss a parallelization architecture for OSNs. To identify the features of OSNs, we collected data from real OSNs, analyzed them, and built a model to describe the behavior of users. Based on this model, we developed methods to predict the behavior of users. According to these predictions, we can schedule our crawler in a more reasonable manner and extract more fresh information using parallelization techniques. Our experimental results demonstrate that the proposed strategies can extract information from OSNs in an efficient and effective manner.

1. Motivation

Online social network (OSN) services are among the hottest new services in recent years and they have vast numbers of users. For instance, Facebook has 874 million active users [1] and Twitter has 500 million users. It has been estimated that at least 2.3 billion tweets were published on Twitter during a 7-month period, i.e., more than 300 million tweets per month [2]. In addition, Yahoo! Firehose receives 750K ratings per day and 150K comments per day [3].

Several social media datasets are available such as Spinn3r [4]. About 30 million articles (50 GB of data) are added to Spinn3r each day, including 20,000 news sources and millions of blogs [2]. People access OSNs frequently, so advertisements can be broadcast according to the behavior of users. [5] studied various Super Bowl ads by applying data mining techniques to Twitter messages. Similarly, using Twitter, [6] detected earthquakes and [7] studied the spread of influenza.

One of the most important factors for crawling is the freshness. OSNs are updated frequently, so old messages will be flushed by new messages within a short period. We define freshness as the average length of the period from when a message is updated on the OSN to when it is collected by our crawler. For instance, we denote a message as fresher if we notice it in one minute after its publication rather than after one hour. Studies such as earthquake detection using Twitter require the latest news and it is much better to detect earthquakes using OSN data obtained in the last few hours instead of previous months. Denev et al. [8] designed a web crawling framework called SHARC, which considers the relationship between the freshness of information

* Corresponding author.
E-mail address: wangzh@hit.edu.cn (H. Wang).

http://dx.doi.org/10.1016/j.future.2015.11.021
0167-739X/

and time. Moreover, Olston and Pandey proposed a crawling strategy optimized for freshness [9], which considers the current time point. OSN crawling is related to web crawling, but crawling fresh information from OSNs is different from web crawling in the following respects, thereby presenting new technical challenges.

1. New messages are published more frequently. Every user can conveniently register, post, comment, and reproduce messages at twitter.com but a server and special skills are required to maintain a website. Thus, the hottest topic may change within a few hours on the OSN. Due to this feature, the freshness metrics for web crawlers cannot be applied perfectly to OSN crawlers.

2. The messages on OSNs are shorter than web pages. The former often comprise a few sentences (e.g., Twitter limits its messages to 140 characters) whereas the latter often contain titles and large amounts of content.

3. OSNs are closely related to the daily lives of users. Ordinary people post messages during the day but not at night. Thus, we must consider work and rest time when crawling for OSN messages. By contrast, this feature is not considered in web crawling.

4. OSN networks are highly complex. The relationships between close users are represented explicitly. Thus, it is easier to trace the friend relationships and forwarded messages on OSNs than on the web. Web pages are usually open to everyone whereas some OSN messages are only available to friends, such as Facebook messages.

Due to these features, new techniques are required for crawling fresh OSN information. Furthermore, many machines run the crawler program at the same time in a real crawler system, which leads to various problems such as task scheduling and workload balancing. Thus, it is necessary to design a parallel crawler architecture.

Given that the goal of OSN information crawling is to gather new information, we aim to crawl the freshest messages possible using limited resources.

2. Contributions

During crawling, the limitations on resources include the bandwidth, computational power, and politeness or etiquette. For instance, at twitter.com, users are permitted to receive at most 200 tweets with a Twitter API call. The restriction on calling the method is 350 calls per hour for one authorized developer account [10]. In fact, this API restriction is the main bottleneck for most OSN crawlers.

To fully utilize the limited resources and to satisfy the freshness requirements for crawlers, we classified users according to their behavior and we also modeled their behavior when updating posts. Using these models, we predicted the post updating times for different users and the crawler could only access the new messages when necessary. As a result, our method can collect the latest information with limited resources.

By combining the steps discussed above, we propose the Crawl based on User Visiting Model (CUVIM) based on our observations and classification of user behaviors. In the present study, we focus on the messages in OSNs. By considering the updating of the relationships between users as a special type of message, the relationships when updating information can also be crawled with the techniques described in this study.

According to the different behaviors employed when updating posts, we classified OSN users into four types: dead account, frequently changing account, reasonably constant account, and authority account. Further details of this classification are given in Section 5. This is the first contribution of this study.

We developed different predication models and efficient crawling strategies. This is the second contribution of this study.

In particular, for dead accounts and frequently changing accounts, the changes can be described by a Poisson process and the rate of change can be predicted by statistical methods. Thus, we built a Poisson model and developed a web crawling strategy to crawl OSN data.

For reasonably constant accounts and authority accounts with frequently posted messages, we observed that the frequency of new messages was related to the daily lives of users. Based on this observation, we found that we could crawl many fresh and useful messages during the day but almost no new messages at night. We built a hash model to visit active users to crawl their information more efficiently.

As a third contribution, we performed extensive experiments to verify the effectiveness and efficiency of the techniques proposed in this study. We crawled the last 2000 messages of 88,799 randomly selected users and the results showed that 80,616,076 messages were collected. According to the experimental results, the Poisson process model collected 12.14% more messages than a round-robin (RR) style method. The hash model collected about 50% more messages than the RR method. The parallelization method limited the difference in the workload of the Poisson process model to less than 13.27% of that for the random method. We also tested our parallelized crawler architecture, where the results showed that the speed up was linear with this architectures while the workload difference between the machines was almost negligible.

The remainder of this paper is organized as follows. Section 2 reviews related work in the area of crawlers. In Section 3, we present our method and our analysis of the data to identify user behaviors. In Section 4, we introduce our crawling metrics. In Section 5, we explain the parallelization technique for the crawler system. In Section 6, we present the experimental results. In Section 7, we give our conclusions and suggestions for further research.

3. Related work

A small number of methods have been proposed for crawling OSN data. [5] described a Twitter crawler developed using Java, where they considered the implementation details for the crawler and the data analysis. By contrast, we focused on the crawling method and we developed algorithms to collect more information from specific OSN users.

TwitterEcho is an open source Twitter crawler developed by Boanjak et al. [11], which employs a centralized, distributed architecture. Cloud computing has also been used for OSN crawling [12], where Noordhuis et al. collected Twitter data and ranked Twitter users with the PageRank algorithm. Attempts have been made to implement parallel crawling, where Duen et al. implemented a parallel eBay crawler in Java and visited 11,716,588 users over 23 days [13]. These three methods aim to perform calculations with more resources, whereas we focused on a more reasonable crawling sequence with limited resources.

Whitelist accounts were once available on Twitter. Kwak et al. crawled the entire Twitter site successfully, including 41.7 million user profiles and 106 million tweets via the Twitter API [14]. However, whitelist accounts are no longer available, as described by [12]. The API is now a rate-limiting process, so we developed algorithms to improve the crawling efficiency.

Another method related to OSN message collection is web crawling, where the changes in pages generally follow a Poisson process model.

The Poisson distribution is a classic discrete probability model in the field of probability theory and statistics [15], which assumes that during a given interval of time, the average rate of occurrence is known for a specific type of event and it is independent of the last time the event occurred. For example, the average number of letters received by a family during one week may be stable and