Research and Implementation of Trusted Data Collection Technology in Network Community

Research and Implementation of Trusted Data Collection Technology in Network Community

Xu Wu [1,2,3]*, Sishu Duan [1,2] and Xiaqing Xie [1,2]

[1]Key Laboratory of Trustworthy Distributed Computing and Service (BUPT)
[2]School of Cyberspace Security, BUPT
[3]Beijing University of Posts and Telecommunications Library
wux@bupt.edu.cn

Abstract. This paper attempts to research and implement the data collection technology and trusted mechanism in network communities, and explore how to improve the reliability and credibility of the data collection process from multiple levels, providing reliable data sources for data analysis.

1. Introduction

Research into and analysis the public opinion of the network community, timely discovery hot topics and tracking the trend of events has become increasingly important [1]. And the timely and accurate analysis results are based on the timely, complete and reliable collection system. In the context of large data, the data of the network community is of large scale, high update speed, deep link level and large noise disturbance, thus requiring more of the data acquisition system. While at present, mainstream web data collection framework has no module to ensure the credibility of the collected data. And there is also no perfect processing logic to process unexpected situation encountered, like the target site revision and program operation exception [2].

2. Related Technology and Demand Analysis

2.1. Data Collection Technology

This paper focuses on the web-based data collection technology with HTTP protocol to collect data[3]. A typical collection program mainly includes four modules, as shown in Fig. 1.The downloader is to obtain the access link, access and download the web page. The information extractor is to extract useful information from the downloaded web page and find out new links. The queue manager is to manage all links. The data persistence module is to persist the extracted and assembled information of the information extraction module[6].

We chose WebMagic as the basic framework, as it is flexible, easy to maintain, of good scalability while providing APIs for further development and customization.

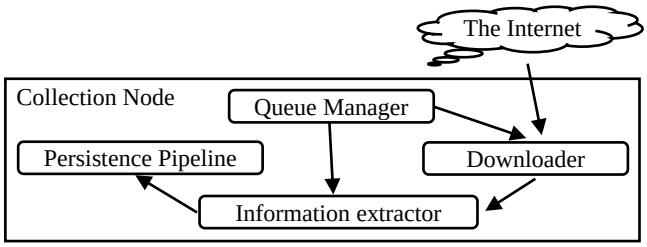

Figure 1. Classic Structure of Web Collection Framework

2.2. Demand Analysis of Credibility

Considering the features of the network community as well as the specific application scenarios of data collection, a trusted technology has to meet four features: sustainability, immediacy, integrity and authenticity [7].

Sustainability. Sustainability refers to the ability of to maintain long-term stable operation confronted with complex situations. Many network communities took measures to prevent automated and massive data collection-requests from outside, and some of them may unforeseeably update their websites, otherwise, the collection tasks may encounter network congestion, etc., these will all lead to a collection failure.

Immediacy. Immediacy refers to the ability of a computer software system to respond to incoming data in a timely manner. In network communities, data update every minute and following hot topics or events demand local data update timely to be analyzed. Considering of the features of these network communities, a reasonable scheme is necessary to improve collection efficiency to ensure the data up-to-date.

Integrity. Integrity refers to the ability of a software system to maintain the overall functionality and integrity of the input and output content. The amount of data and update frequency differs among different communities, without dynamically adjustment according to the traffic; it may leak data when the amount of data is large or cause a waste of performance when that is small. In order to collect data timely and completely, a mechanism for monitoring or verifying the integrity of data is necessary.

Authenticity. Authenticity refers to the ability of the system to ensure that the results of functions are true and reliable. The content in these websites were created by users and may be revised or deleted at any time, which happened all the time. To maintain the original records to be traced back, it is necessary to have a certain mechanism for data collection framework to verify the collected data, timely access to and capture the content modified or deleted by the user [7-10].

3. Design of Trusted Data Collection Mechanism

The trusted data collection mechanism (TDCM) in this paper include four parts of the source sustainability, instant collection, dynamic management and reality validation, trying to improve the immediacy, the integrity, immediacy and authenticity of the collected data in multiple levels, as shown in Fig.2.

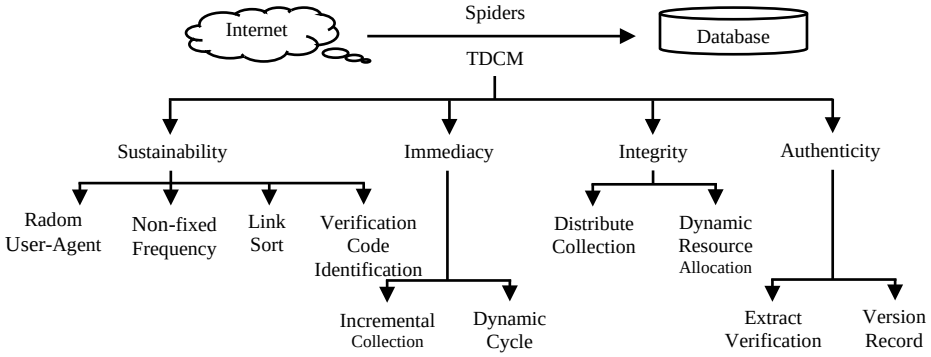

Figure 2. Architectural Design of the Trusted Data Collection Mechanism

Sustainability. Many websites take measures to prevent massive automated requests. Some websites can utilize the field "User-Agent" in a request header of the HTTP request, and they can also use the image verification code or the frequency of the request to determine whether it is an automated collection program.

In order to ensure sustainable collection, we design the four methods:

1)Randomly using multiple commonly 'User-Agent' values;

2)Sleep between the requests for a period of random time to make sure the request frequency is not fixed;

3)Randomly sorting the links needs to be request further to make sure the order of link requests is random and unpredictable;

4)Automatic identify the verification code and train the accumulated results, improving the recognition accuracy.

Immediacy. Incremental collection technology that is recorded the location of this collection after each process completed, and each collection only collected the new content produced after the last collection cycle, as shown in Figure 3.

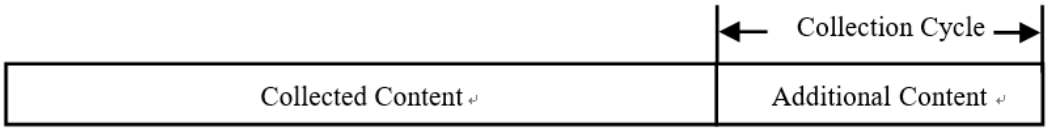

Figure 3. Incremental collection diagram

The collection program collects data of the network community at a certain cycle, and reads the preset cycle value from the scheduling module each time it starts. After a cycle is finished, the next collection cycle is dynamically adjusted according to the ratio of actual operation time and the preset cycle time.

When there is a sudden increase or decrease of the new release contents, the system will dynamically change the cycle time of the nodes to collect new content timely.

Integrity.To handle massive data, we design an independent queue from the collection nodes, the queue management tools can respond to the high concurrency to manage the nodes, forming a multi-node collaborative and parallel collection structure and improved the collection system structure. As shown in Fig.4. multiple identical collection nodes run on different machines, access links from a unique queue manager, and then collect, clean and store data in parallel.

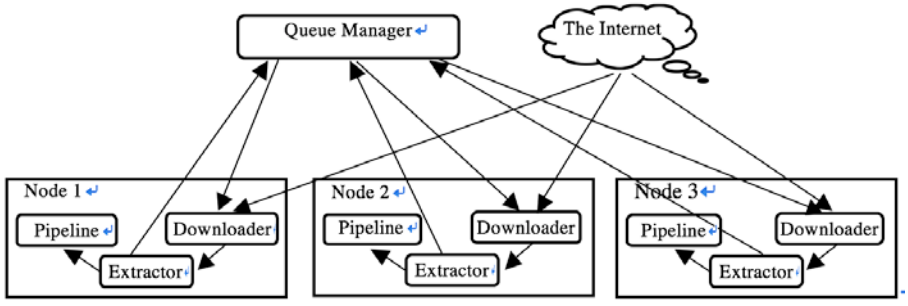

Figure 4. Distributed parallel collection node structure diagram

Authenticity. To improve authenticity, independent trusted authentication module is designed to collect the modified or deleted content in time by validating the authenticity of the data over a period of time, we extract the data collected in a number of days in each cycle (like 3 days) and randomly pick other data, then re-visit their addresses to detect whether the data has been modified or deleted.

For the demands of keeping original data like data snapshot, we design a separate version table to store the modified or deleted article, while the latest version stored in current table. In this method, we can retrieve all historical versions.

4. Implementation of Trusted Data Collection Mechanism

As mentioned before, this paper chosen WebMagic as the basic framework.

4.1. Implementation of Sustainability

Select a website with anti-collection policy and take measures to bypass the policy:

1)Encapsulates the upcoming request in Downloader, and automatically set the User-Agent field in the Site object passed from PageProcessor to one of the commonly used browser. And then set the Site object's properties and parameters to the HTTP request, preventing from being recognized as an automated program.

2)Customize the Downloader. Sleep a random time between requests sent by collection program so that the frequency of a single collection node is not fixed.

3)Customize the queue manager (Scheduler). Each time put one link into the team, call the Collections.shuffle () method for random sorting.

4)Implement verification codes recognition. Simple codes can be settled by image segmentation comparison, Complex ones can be identified by OCR (Optical Character Recognition) technology, one commonly used tool is Tesseract-OCR. Encapsulates the tools and logic of verification codes recognition, we will be able to identify the image verification code.

4.2. Implementation of Immediacy

Firstly, an incremental collection logic is to timely capture only the new content on the network community without traversing the whole site every time.

According to the typical structure of "List - Detail" page, the post’s id in one board is incremented. When the cycle begins, the list page is judged whether the id of every post is bigger than the maximum id saved in last cycle. Otherwise, it will be skipped.

Secondly, a mechanism of dynamical cycle time is applied. When there is a sudden increase or decrease of traffic, the system will dynamically change the cycle time of the nodes to collect new content timely.

In implementation, extend a layer of SpiderManager over the Spider object, use the Timer and TimerTask tool to perform periodic task management on collection tasks. The scheduling module will determine the need to modify the task cycle time at the end of each cycle. If yes, it will call timer.cancle() method to cancel the existing task, and then call the timer.schedule (task, cycleTime) method to reboot a new cycle.

The program sets a minimum and maximum trigger period (for example, 1/3 and 2/3) in the dynamic cycle. If the actual running time is less than 1/3 of the setted running cycle, the program judges that the amount of current data is smaller and will then shorten the cycle period. Otherwise in situation of more than 2/3, cycle period needs to be longer to prevent data leakage.

4.3. Implementation of Integrity

We deployed the framework in distributed architecture, separate the queue manager of the collection node and use the queue management tool.

Firstly, inherit the default QueueScheduler and modify the access and removal interface into a remote centralized queue manager. In this structure, collection processes of one website run on different machines. For websites with a large amount of concurrent data, the queue management module will be separated from the shared queue management tool so that the collection task will operate in parallel.

Secondly, implement the dynamic management of the distributed framework. Control procedures operate respectively in the master server and slave server, communicating by socket. Then slave server calls Runtime. getRuntime().exec() method to start the collect program, it will monitor report running state,including local memory, CPU usage and other information. The scheduling module will change the number the collection nodes accordingly.

4.4. Implementation of Authenticity

Authenticity verification runs as a standalone task every day, it uses SQL statements to get posts

within several days (for example, three days) from MySQL, and randomly picks posts of other time. Then get their address to re-visit through their site name, board id, threads id and article id to detect if the data is modified or deleted. The results will be stored in the verification results database.
And also, we use a separate version table in database for each item that needs to be validated, and the tables recorded the revised records of board, article and user table respectively while the latest content saved in current table.

5. Verification of Trusted Data Collection Mechanism

5.1. Verification of the Sustainability

In each cycle, when a collection request fails it will retry two times, and if all in failure, it will be marked as failure. The retry rate and failure rate are defined as Equation 1 and Equation 2, *s* presents the number of successful pages, *r* represents the number of retries for all pages, *f* for the number of failed pages.

$$\text{Retry Rate:} \quad rRate = r / (s + r + f) \tag{1}$$

$$\text{Failure Rate:} \quad fRate = f / (s + r + f) \tag{2}$$

Take a college community with anti-collection technologies as an example, the running results in 12 cycles are shown in left picture in Fig.5.

With the trusted module, retry rate is at about 1.5%. In every 10 requests, they typically retry 10-20 times. Most of them are successful when retry, the failure rate remains below 0.05%. The module effectively enhances sustainability.

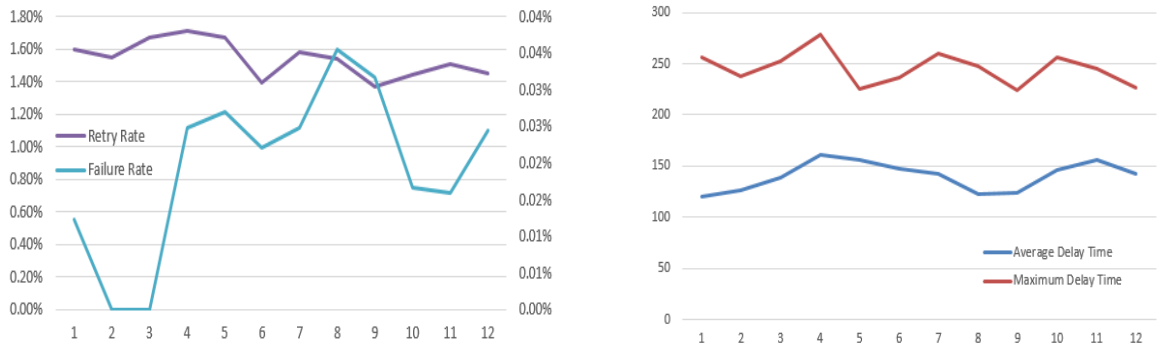

Figure 5. left:Sustainability Verification Results; right: Immediacy Verification Results

5.2. Verification of the Sustainability

We use the Delay Time to measure the immediacy, in Equation 3 and 4, *pt* represents the post time, *pc* represents the collecting time.

$$\text{Delay Time:} \quad lt = pc - pt \tag{3}$$

$$\text{Average Delay Time:} \quad at = (n \sum lt) / n \tag{4}$$

Select a website with more than 100,000 daily posts to verify the effect of incremental collection. Program without the incremental function runs about 22.5 hours each time to traverse it over, which is also the Delay Time. While for the program with incremental collection and dynamic cycle function, as shown in the right picture in Fig.5, the average delay is stable at about 120 seconds, and the maximum delay is no more than five minutes. The improvement effect is obvious.

5.3. Verification of the Integrity

For the verification of node scheduling, we select a website with more than 200,000 daily posts to verify the integrity. For the stand-alone version, data cannot be collected completely even when the node at full load. For the program with distributed node scheduling function, we record the stats of 24-hour distributed collection.

In the time period when traffic increases (8:00 - 09:00), the node scheduling module identified the increase and dynamically add nodes for data collection. When the traffic reduced (00:00 - 3:00), the node still recognized the decrease and dynamically reduce the number of nodes, as shown in Fig. 6.

For the verification of Board Monitoring**.** After the collection program ends, manually modify the database by removing or adding one of the board data to simulate the situation. The results of the experiment show that the board monitoring module can correctly identify the change of the boards. It can effectively prevent the data leakage caused by the lack of timely identification of changes, improving the integrity of the collected data.

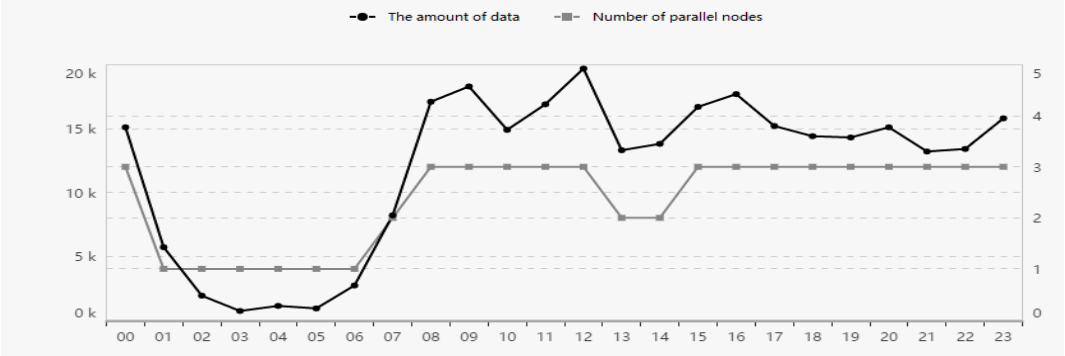

Figure 6. Distributed Collection Framework Node Configuration Diagram

5.4. Verification of the Authenticity

To verify the function of the verification module, we compared the latest content with the contents in database one by one. In Equation 5, *c* represents the number of all contents in database, r *c* represents the number of same data as rc. We select a very active network community to verify data during a month, results are shown in Table 1.

The experiment shows the posts modified or deleted in three days can be identified correctly and the historical contents are all be collected and marked successfully. And the real rate of the posts within a week and a month are above 99.99%.

$$\text{Real Rate of Data: } aRate = rc / c \tag{5}$$

Table 1. The authenticity verification results of the collected data

	Posts in 3 days	Posts in a week	Posts in a month
Total post count	28754	206436	852384
Real post count	28754	206434	852323
Real Rate	100%	99.9990312%	99.9928436%

6 Conclusion

This paper designed and implemented trusted mechanism in four aspects, sustainability, immediacy, integrity and authenticity. Experiments show that the framework can provide better credibility in the sustained and stable operation at the same time. Still, there are room for further improvements, for example, we improve the automation of the collection framework to adapt to different types of websites.

7. Acknowledgements
This work is supported by National Key Research and Development Plan (No.2017YFC0820603), Director's Project Fund of Key Laboratory of Trustworthy Distributed Computing and Service (BUPT),Ministry of Education (No. 2017ZR02).

8. References
[1] Venkatesh, V., Croteau A.M., Rabah J.: Perceptions of effectiveness of instructional uses of technology in higher education in an era of Web 2.0. In: 47th Hawaii International Conference on System Sciences, pp:110-119. IEEE Press, Hawaii (2014).
[2] Srivastava J., Cooley R., Deshpande M.: Web usage mining: Discovery and applications of usage patterns from web data.J. Acm Sigkdd Explorations Newsletter, 1(2): 12-23(2000).
[3] Kausar M.A., Dhaka V.S., Singh S.K.: Web crawler: a review.J. International Journal of Computer Applications, 2013(2): 42-45.
[4] Brawer S.B., Ibel M., Keller R.M.: Web crawler scheduler that utilizes sitemaps from websites: U.S. Patent 9,002,819[P]. 2015-4-7.
[5] Qian R., Zhang K., Zhao G.: A topic-specific Web crawler based on content and structure mining.C. In:3rd International Conference on Computer Science and Network Technology (ICCSNT). IEEE, 2013: 458-461.
[6] Farooqui M.F., Beg M.R., Rafiq M.Q.: Validation of Architecture of Migrating Parallel Web Crawler using Finite State Machine[J]. 2014: 420-439.
[7] Tao H.W., Chen Y.X.: A metric model for trustworthiness of software.J. Proc. of the 2009 IEEE/WIC/ACM Int'l Conf. on Web Intelligence and Intelligent Agent Technology. 2009: 69−72.
[8] Liu Y.Z., Luo X., Xue K., Luo P.: A metric model research based on attributes for trustworthiness of software.J. Computer Science and Application, 2012(2): 121−125.
[9] Zhang L.W., Zhou Y., Chen Y.X.: Stability of software trustworthiness measurements models.J. Proc. of the 7th Int'l Conf. on Software Security and Reliability Companion. 219−224(2013).
[10] Pedraza-Garcia G, Astudillo H, Correal D. Modeling Software Architecture Process with a Decision-Making Approach.C. In 33rd International Conference of the Chilean Computer Science Society (SCCC),. IEEE press, 2014: 1-6(2014).