Comparative Analysis Before and After Optimization of URL Detection Program

Yi Yan*, Lichao Zhan*, Susheng Zhongb, Xijun Linc, Xiaohui Maid and Kaidong Zhoue

Information Center Guangdong Power Grid Company Limited, Guangdong, China

*Corresponding author e-mail: yanyi@gdxx.csg.cn, azlc4work@163.com, zhongsusheng@gdxx.csg.cn, 827860684@qq.com, 13678990532@139.com, 13924230718@qq.com

Abstract. Perform URL detection on the three major systems of the power grid, and obtain the program for the detection results. The whole process of the project has gone through two stages, namely the original program construction and the optimized program construction. The original program was built in a cluster with 918 menus and 36 ips and ports. Since there is not a proper number of threads, a large number of URLs are waiting in serial during URL detection; the final data is exported to excel, resulting in The detection of the cluster resources is wasted, the total length of the probe is about 35 minutes; the optimized program adds enough threads, and the data is exported to the excel mode, the data is exported to the log file through the log framework, and finally A total of 2 minutes, the efficiency increased by 1650%.

1. Introduction
With the continuous development of the power grid system, attracting more and more customers, it is very important to ensure the customer's satisfaction when the customer accesses the grid system, so that it is very important to improve customer satisfaction. First, we will perform http request test on all URL links of the webpage after a certain period of time, and obtain the failure codes 404 and 500 of the successful 200 or failure, and then, after obtaining the data, perform corresponding two accesses to the URL that fails to access. To determine three cases, the first type of access determines whether the URL is enabled, the second type of access determines whether the network is normal, and finally, if the previous two visits are normal, then it can be determined that the access failure should be the system hardware, running The compatibility and other issues, and then the corresponding algorithm model to locate the corresponding fault. The three steps mentioned above constitute the process of detecting, locating and predicting the entire URL, in which the http request test is performed on all URL links of the webpage, that is, the process of URL detection is the basic data acquisition process, and is also an important process. The next two steps provide important data sources. The URL detection platform also encounters bottlenecks in the construction process. Resolving these bottlenecks and reducing the overall detection time facilitates quickly access to access fault data and saves a lot of time and cost.
2. URL probe infrastructure construction
The URL probe completes all URL links to all ip and port menus of all three major systems in the grid cluster. The specific process is as follows:
1. The program starts, creating a timing task to implement the detection task
2. Get the URL of the menu of all menus in the marketing system, customer service system, and personalization system.
3. Traverse the queried URL menu array, spell all the different ip and port numbers under different system clusters, and probe them according to the created thread pool allocation thread.
4. If the thread has an http request exception in the probe, it will output an exception message in the probe exception log.
5. Wait for all threads to complete the request task, that is, complete a URL probe process
6. Return to the first step and execute the process here according to the requirements.
The summary is as follows:

![URL detection process diagram]

Figure 1. URL detection process

3. Grid system cluster
The grid system platform not only satisfies its own marketing records, query requirements, customer consulting and after-sales service requirements, but also integrates a huge cluster of individualized needs. These requirements face the three major systems of the grid system: marketing system, customer service system, personalized system; these requirements are connected to the corresponding page by a number of URLs, which means that each system will correspond to multiple ips and ports, of which the sales system menu occupies an overall 64.3%, ip and The number of port numbers occupies an overall 55.6%. The ip and port number information of the three major systems of the grid system are given below, as shown in Table 1:

| System                  | IP1                          | Port number 1, Port number 2, ... |
|--------------------------|------------------------------|----------------------------------|
| Marketing system         | IP1                          | Port number 1, Port number 2, ... |
|                          | IP2                          | Port number 1, Port number 2, ... |
|                          | ...                          | ...                              |
| Customer service system  | IP1                          | Port number 1, Port number 2, ... |
|                          | ...                          | ...                              |
| Personalized system      | IP1                          | Port number 1, Port number 2, ... |
|                          | ...                          | ...                              |
|                          | 590 menus, 20 different ip and port numbers |
|                          | 170 menus, 8 different ip and port numbers |
|                          | 158 menus, 8 different ip and port numbers |
4. Initial program built

4.1. Process
After determining the overall architecture of the grid system and the number of probes to be detected, a preliminary procedure is established. In the initial procedure, one-by-one detection is adopted, that is, an ip corresponds to the previous port number, and the ip and port number are performed. Detection of all URLs, detecting the status code of the access of the URL under each menu bar, filtering out the menu of access failure, exporting it to the excel table for subsequent positioning prediction research, and then opening another one of the same ip. The port continues to detect, and the result of the access failure menu is exported to excel, and all the menus under all port numbers under all ips are repeatedly executed in sequence, and the flow is summarized as follows:

Figure 2. Initial program setup process

4.2. Result
The total number of accesses required for the entire grid system to be accessed is the number of menus per system multiplied by the sum of the ip plus the port number, so the total number of visits obtained is 14,424, and the initial program built the probe for each port number menu. It takes about 2 minutes, so it takes 35 minutes to probe one by one, and our requirement is to have all the addresses in 5 minutes, and 35 minutes of long-term detection will cause time and cluster resources to be wasted. Therefore, it is necessary to change the initial program built and optimize the program.

5. Optimized program built
For the analysis of the three places that take a long time, the initial program setup only opens a probe program. After the program processes all the addresses of one system, it will detect the next system, which will cause the program to run. OK, it takes time to wait, the optimized program built starts three programs at the beginning, corresponding to detecting three systems, then the number of threads per program, the number of threads in the beginning is not enough. This causes a large amount of waste in the waiting time caused by the URL not being detected. Therefore, in the optimized program construction, each thread opens enough threads to ensure that it can be completed in a short time, for example, in the detection marketing. When the system starts, the thread pool of 2000 threads is opened. Finally, in the initial program setup, we export the error code of the access error to the excel table. Since the data is exported to the excel table, the data needs to be first placed in the memory. The data in memory is written to excel, which will increase the IO pressure of the cluster, and also consider the thread safety problem of processing data on multiple threads. Therefore, the overall efficiency will be reduced. Therefore, in the optimized program construction, we adopt the slf4j log framework. When a single thread executes a URL detection task, if there is an exception during the detection, the URL information of the detection exception will be recorded in real time and output. To the exception log, since the output to the log is real-time, the efficiency is greatly improved. Table 2 below shows the initial program built improvements:
### Table 2. Program Improvement

| Number of programs | Initial program setup | Optimizer built |
|--------------------|-----------------------|-----------------|
| Number of threads  | Multithreading under a single port | 2000 or 300 |
| Data output        | Excel                 |Slf4j log frame  |

### 5.1. Optimizer process

Enable three probe programs, each probe detects the URL address of a different system, where each program opens a correspondingly sufficient number of threads to ensure that the probe is completed in a short time without causing cluster resources. Waste, the URL address is no longer an ip corresponding to a port number, the serial access of all URLs under this ip and port number is detected, but the different ips of the system and different port numbers are combined. The menu is placed under the thread pool for detection. The slf4j log framework used in the process of detection will unify the accessed information into the same log file every day. Figure 3 shows the flow chart of the optimized program:

![Figure 3. Optimizer flow chart](image)

### 5.2. Results

The number of accesses required for access to the entire grid system is 14,424. The optimized program accesses the full URL address to the log output in a total of 2 minutes, greatly improving the detection efficiency. The comparison results are shown in Table 3 below:

### Table 3. Comparison results

| program | Initial procedure | Optimizer |
|---------|-------------------|-----------|
| time consuming | 35min | 2min |
| Promotion | - | 1650% |

### References

[1] Chen Yi, Wang Pei. Application of Synchronization Mechanism to Solve Java Multi-thread Security Problem [J]. Software Guide, 2018, 17 (12): 165-168+172.

[2] Sun Shuai, Li Chuandi. Analysis and Implementation of Multithreading Management Based on Linux [J]. Electronic Science and Technology, 2017, 30 (09): 27-29.

[3] Wu Zhifeng, Pei Fengjuan. Application Research of SLF4J Framework Integrated Log System [J]. Information Communication, 2011 (04): 74-75.