Design and Research of Batch Query System Based on Java

Menglin Liu[1], Li Zhang[1], Zhenzhen Wang[1], Jianguang Liu[2]

[1]College of Information Science and Engineering, Zaozhuang University, Zaozhuang, Shandong, 277100, China

[2]Zaozhuang Unicom Company, Zaozhuang, Shandong, 277100, China

Keywords: Java, Batch query system, Database

Abstract: The query for detailed call records is an important window of Telecom operators to provide services for the outside world. For Shandong Unicom Corporation, as current query system of detailed call records can only achieve the query of a single number or batch numbers in a single account period, but it cannot achieve the query such as batch number in a multi-account period. In order to solve this problem, an optimization plan of batch query is proposed and the query system with a graphical user interface is designed by using Java technology. Practice has proved that the query system is easy to operate, and it has the advantages of high query efficiency and accurate query results, so it has certain application and promotion value.

1. Introduction

In daily life, people often interconnected via telephone or SMS. Data generated by these ways will be stored to the appropriate database in real-time. When the public security organs, procuratorate, court and other units or individuals (hereinafter referred to as users) need detailed call records about some specified phone within certain stage to handle business, each telecommunications company should extract data from these databases for them accurately and quickly, therefore, for the telecommunications companies ,the query for detailed call records is an important window to provide services for the outside world. To this window, Quality of Service will directly affect the company's image. for example, In Shandong Unicom Corporation (Hereinafter referred to as the Company), information service center often need to provide detailed call records about voice and SMS of some specific Unicom mobile phone number or landline number within certain stage for the public security organs, procuratorate, court and other units or individuals. Although the current query system of detailed call records is running well during single number is queried, but it has the following problems: query efficiency is low, query results are stored cumbersome, and running process is influenced by parallel transactions. These problems has affected the service quality of Unicom seriously.

In addition, the company usually carry on investigation for the user's communication habits and call size, Bulk detailed list has become the main research material. So batch detail query is the main means of the company's traffic research, and optimized research of batch query for detailed call records is imperative.

In order to solve the question of batch query for detailed call records in multiple accounts, In this paper, on the basis of full investigation, use the SQLyog client to create the middle table and stored procedures in the call list database, according to the batch queries, call statements are generated. Then using these call statements to call stored procedure, the query results are stored in the middle of the table, then the query results are extracted from the middle table. So the batch query are finished .

In order to facilitate the operation of the operator, the team used the Swing graphical user interface technology, JDBC technology and I / O streaming technology provided by Java for the system design a set of beautiful, easy-to-use graphical interface. In this way, even if the staff do not understand the database ,he also can quickly and accurately complete the call detail batch query.

2. The Status of Current Query System

2.1 Low query efficiency

The function about Current query system of detailed call records is relatively simple.This system only can query a single number or batch numbers in a single account period at once. and the query is not efficient. If you want to query a number of call details within a year, you will query 12 times, each query needs about three minutes, these queres will take about 3 * 12 = 36 minutes. And so on ,If you want to query for batch numbers within many account periods, so consuming long can imagine.

2.2 Cumbersome Query result storage

Using the Current query system of detailed call records , the query results are stored cumbersome. For each number within each account period, each query result need to be preserved by oneself. If you want to query detailed call records for 10 numbers in 12 months, you will have to produce 10 * 12 = 120 files, so many files to be stored will cause storage cumbersome, efficiency decreases and poor maintenance. And so on ,If you want to query for batch numbers within many account periods, the stored query results will be great. The workload to store query results will be enormous.

2.3 The impact of parallel transactions

Because the current query system of detailed call records is a small part function of ESS and the parallel transaction is more in ESS, so its speed is affected by other parallel transactions greatly. If there is a large amount of parallel transactions , the running speed of the current query system will become very slow.

2.4 Inconvenient to operate

The current query system does not provide a visual interface, it needs professional technical staff who understand the database to complete the query function, which greatly limits the scope of its use.

3. Design Research Program of Query System

In order to achieve batch query for batch number in the multi-billing period and to meet the fast, efficient and accurate call detail inquiry needs, the panel members have put forward the following optimization program by analyzing the problems of current query system of detailed call records:

The overall program: make full use of the historical detailed database, select and install Mysql's client: SQLyog whitch can running fast, create the intermediate tables and stored procedures in this database, call to these stored procedures, quickly and accurately generate and extract query results.

Firstly: According to the needs, five intermediate tables are designed using to store data: query requirements data that is user-submitted, mobile voice data, mobile SMS data , Fixed telephone voice data and batch query generation data, which can solve the problem that query results are stored too many.

Secondly: Three stored procedures are designed using to check out the detailed call records from the database, and the data are inserted into the following tables: one table to store the mobile voice data, one table to store the mobile SMS data and one table to store Fixed telephone voice data. Using Java Web technology, graphical user interface is designed and is used to Export query results , obtain the final detailed call records, which can improve the query efficiency.

Thirdly: Because SQLyog runs fast, and it is affected by other systems, so take advantage of SQLyog to solve the problem on the local machine by other concurrent transactions influence, and improve the operating speed.

Fourthly: In order to facilitate the operation of the operator, the team used Java language to design a lot of detailed inquiry system graphical user interface. Through the operation of the localization interface, it is convenient to batch detailed query and generate query results automatically to improve the related work efficiency and information service level.

4. Implementation Process of Query System

4.1 Create intermediate tables

One table is used to store demand. The structure of this table is following:

zz_haoma (haoma varchar2(20), type_cx varchar2(1), start_yymm date, ds_mounth number, cxr varchar2(20));

This is the detailed query requirements table, which is used to store the numbers of batch query, query type, start date and person’s name who makes a query request.

The other one table is used to store the batch call statements.

The structure of this table is following:

zz(v_v varchar2(1000), s_t varchar2(2));

The other three tables will be used to store mobile voice data, fixed telephone voice data and mobile SMS data.

among them:

sd_zaozhuang.ucr_zz_tg_cdr10: used to store the bulk of the query to the mobile voice detail data;

sd_zaozhuang. ucr_zza_tg_sm :used to store the bulk of the query to the detailed sms data;

sd_zaozhuang. ucr_zza_tg_cdr10_fix: used to store the bulk query to the fixed-line voice detailed data;

4.2 Create Stored procedure

According to the table zz_haoma , the stored procedure ” plxd_shch” is created which can generate the statement of batch call.

for example: telephone number 18606420981 is stored in the table of zz_haoma ,if you will generate batch detailed call statements from September 2016 to March 2017,you can call : plxd_shch (1,2,201509).Call statement is generated as follows:

call sd_zaozhuang.cdr_insert('186.......1','20151101000000','20161130235959','0632','0');
call sd_zaozhuang.cdr_insert('186.......1','20151201000000','20161231235959','0632','0');
call sd_zaozhuang.cdr_insert('186.......1','20160101000000','20170131235959','0632','0');
call sd_zaozhuang.cdr_insert('186.......1','20160201000000','20170229235959','0632','0');
call sd_zaozhuang.cdr_insert('186.......1','20160301000000','20170331235959','0632','0');

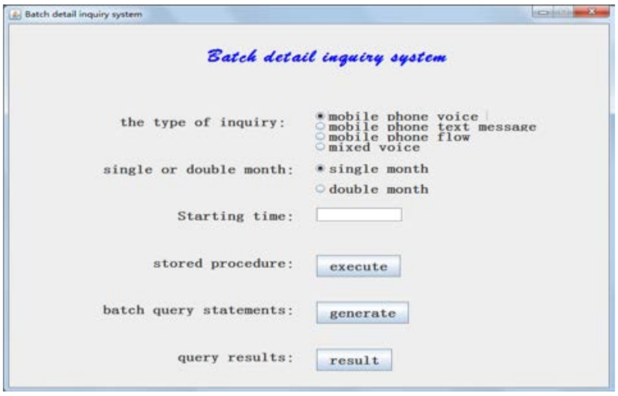
In addition, we have to create the following three stored procedures in the company history detail database:

sd_zaozhuang.cdr_insert:it is used to query the details of mobile voice data and insert into the intermediate table sd_zaozhuang.ucr_zz_tg_cdr10.

sd_zaozhuang.cdr_sm_insert: it is used to query the details of mobile sms data and insert into the intermediate table sd_zaozhuang. ucr_zza_tg_sm.

sd_zaozhuang.cdr_fix_insert: it is used to query the details of fixed telephone voice data and insert into the intermediate table sd_zaozhuang. ucr_zza_tg_cdr10_fix.

4.3 Design and Implementation of Graphical User Interface



Batch detail inquiry system

Batch detail inquiry system

the type of inquiry: ● mobile phone voice ○ mobile phone text message ○ mobile phone flow ○ mixed voice

single or double month: ● single month ○ double month

Starting time:

Figure 1 graphical interface of Query System

In order to facilitate the operation of the operator, the team used the Swing graphical user interface technology, JDBC technology and I / O streaming technology provided by Java for the system design a set of beautiful, easy-to-use graphical interface. The graphical interface as shown in Figure 1. In this way, even if the staff do not understand the database ,he also can quickly and accurately complete the call detail batch query.

The core code is as follows:

```
public void zzxcPrPlxd_shch(int typeId,int month,String cyle) {
        try {Class.forName("oracle.jdbc.driver.OracleDriver");
              Connection conn = DriverManager.getConnection(
              "jdbc:oracle:thin:@ 134.45.8.49:1521:ZZJF", "zhuq", "bill2012");
              System.out.println("lianjie chenggong ");
              String callsql="{call PLXD_SHCH(?,?,?)}";
              CallableStatement csmt=conn.prepareCall(callsql);
              csmt.setInt(1,typeId);
              csmt.setInt(2,month);
               csmt.setString(3,cyle);
              csmt.execute();
              csmt.close();
              conn.close();
              } catch (Exception e) {
        }
  }
```

4.4 Get the Result of the Query and Submit It to the User

When we click "result" button in Figure 1, the query system in the company bill compression system one by one batch call statement, according to the type of query has been selected, the selective implementation of the stored procedure UCR_ZZA_TG_CDR10_FIX, UCR_ZZA_TG_SM, UCR_ZZA_TG_CDR10, UCR_TG_GS query results Insert into the corresponding tables of UCR_ZZA_TG_CDR10_FIX, UCR_ZZA_TG_SM, UCR_TG_GS, UCR_ZZA_TG_CDR10, and then inquire the views GH_SELECT, YDSM_SELECTYD_SELECT and GS_SELECT, and finally generate the query result and save it to the local disk.

5. Optimization results and analysis

After the stage of testing and evaluation, the batch number of multi-billing period Detailed query optimization is now on-line and put into use. During this period ,a larger typical demand is: query 21,080 mobile phones from September 2016 to June 2017 for ten consecutive months of detailed list for today's mobile phone market research. If we use the current query system, the workload is enormous, even it is almost impossible to be completed. But we use the optimization system to query the detailed data quickly. The final results and related indicators as shown in Table 1.

Table 1 the Optimization analysis of batch query

Optimization metric name	unit	Optimization metric	
The number of demand	Number	21080	
the number of accounts	month	10	
Generates the number of call statements	item	210800	21080*10
Generate call statements time consuming	second	20.481	
Batch calling takes time	second	596900	
The number of queries	item	1928841	
Average time-consuming	second / item	2.831	

From Table 1, we can see that the optimization of the batch query is realized by the combination of batch detail and multiple accounts, the query process is simple, the batch call speed is fast, the query result is accurate.

6. Conclusions

This optimization of Batch Inquiry for Call Details is a useful complement and innovation for the company's detailed inquiry system. It is the result of paying attention to full research and seeking truth from facts.

After nearly a year of use, practice has proved that optimized batch query system provides a more efficient, more accurate and more efficient services, and can well meet query needs for the user and the company. The optimization has some practical value.

References

[1] Bruce Eckel. thinking in java (Fourth Edition). Machinery Industry Press . 2007

[2] Yi Luo, Wei Wang, Xuemin Lin, Xiaofang Zhou, Jianmin Wang, Kequi Li.SPARK2: Top-k Keyword Query in Relational Databases. IEEE Transactions on Knowledge and Data Engineering . 2011

[3] Jianhua Feng, Guoliang Li, Jianyong Wa. Finding Top-k Answers in Keyword Search over Relational Databases Using Tuple Units. IEEE Transactions on Knowledge and Data Engineering . 2011

[4] P. He and Y. Liu: Electronic Technology & Software Engineering, 2014 No.5, p.42-43.(In Chinese)

[5] Paul DuBois: MySQL technology insider , 2015. (In Chinese)