Design and Implementation of Automatic Astronomical Spectra Data Processing System

MENG Fan-Long[1,a], PAN Jing-Chang[2,b]*

[1,2]Shandong University, Weihai. 264209, China

[a]email:morphy8@163.com, [b]*email:pjc@sdu.edu.cn

Keywords: Log Storage; Sky Information Storage; Automatic Storage; HttpClient

Abstract. Astronomical observation produces a large amount of data. The effective storage of the data is an important work. In this paper, the automatic processing of this kind of data was studied in order to solve this problem. The log file, the personnel on duty manual record data and information in exposure are integrated to display and process in a single interface. It makes the sky region information storage and spectrum information storage simple. Based on actual demand we designed the automatic storage system. Instead of manually putting the log into storage, users now can modify the contents of the automatic storage information by checking logged data to ensure correct information. Actual practice indicates that this system improves the efficiency of data processing greatly.

Introduction

Astronomical observation produces a large amount of data every day. We must organize these data together and save them in databases in order to process and publish them for further use. We study the method of log storage in this paper. The log information consists of two parts, one is the sky information and the other one is the spectrometer information. The whole sky is divided into many regions in order to be identified easily, each region simply being called a sky. The information of a sky contains the identified number, phase of the moon, exposure time, seeing, center RA, center Dec, etc. These information needs staff in NAOC (National Astronomical Observatories, Chinese Academy of Sciences) integrates the log file and the exposure data, then inserts them into databases one by one. Observation data is always very large and the processing is difficult. Also manual operation is inefficient and error-prone. In this paper we make the operation easier and more accurate by automatic process of the observation data. We design the automatic log storage system in order to put the data into databases based on the exposure data and log file automatically instead of tedious manual operation. The system based on java + hibernate + bootstrap + jquery technology in B/S[1][2] architecture, a good solution to solve the portability problems.

Module Design

The log storage system consists of three parts: logged, unlogged, and auto-storage. The module “unlogged” displays the list of dates in which the log has not been put into databases. The module “logged” shows the dates in which the logs have been saved. The logs have been put into the database already and those have not been saved are all divided into groups by year in reverse order. The third part “auto-storage” can save the observed information automatically in a date range. The system modules chart is shown as Fig.1.

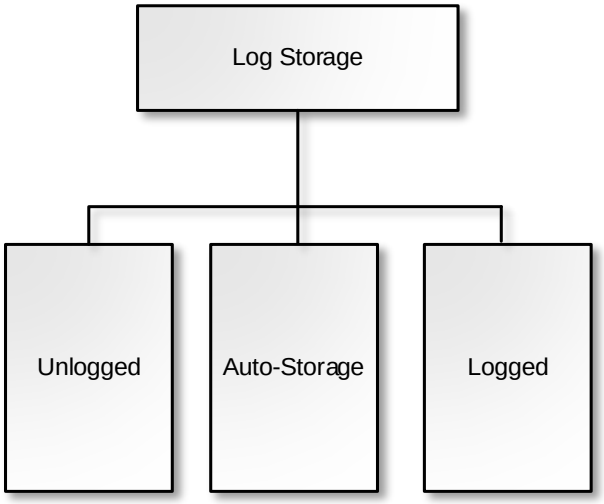

Fig.1. System Modules Chart

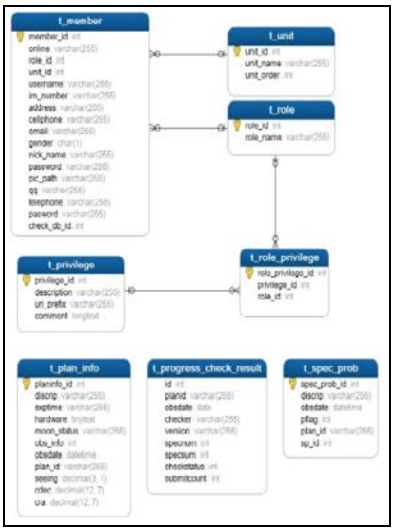

Fig.2. Database Design Diagram

Database Design

As a subsystem of the “LAMOST Product Process Monitoring System”, “Log Storage System” uses two databases, one of them is lasacv2 which is created by ourselves locally and the other is dbregular2 which is provided by NAOC, and there are eight tables altogether. As shown in Fig.2.

Operation Procedure

When users carry out the operation “store log”, the system will only save the log information into the database “lasacv2” instead of “dbregular2”. The operation “Update Lamost DB” need executed to make the two databases synchronized. In the background of this operation, the system finds the different between the two databases and copies the newest data from “lasacv2” to “Lamost DB”.

Unlogged Process

The page “unstore log” lists the dates in reverse order in which the log information has not been saved, as shown in Fig.3.In this page, there are two choice for users to select, on the one hand, users can enter the page “fill log information” by clicking the hyperlink of the date and on the other hand, users can update the database in NAOC (“dbregular2”) by clicking the button “update lamost db”. In the “fill log information”, we can see the observation information, including log file information (Fig.4), information recorded in “Collaborative Work Platform for LAMOST Network”(Fig.5) and exposure information(Fig.6). Before the information displayed in the page, we deal with the information in the log file and exposure table. A sky in a day may make multiple exposures and every exposure produces a “seeing” value and its exposure time may be different. We need to calculate the average of the “seeing” and sum up the exposure time according to the sky number.

When the page “fill log information” initialized, the system lists several records of skies information based on treated exposure table. The skies information, such as “seeing”, “exposure time”, “cra”, and “cdec” are filled out automatically according to the treated exposure table. As to the phase of the moon, the system calculates the lunar calendar corresponding to the calendar in the background and fill it in the drop-down list. With the help of auto-storage the information users need to fill out is observation type, hardware condition and observation description, as shown in Fig.7.

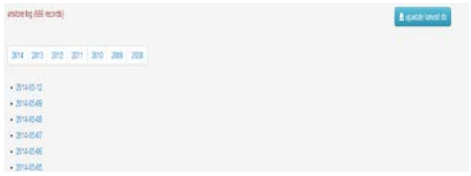

Fig.3. unstore log

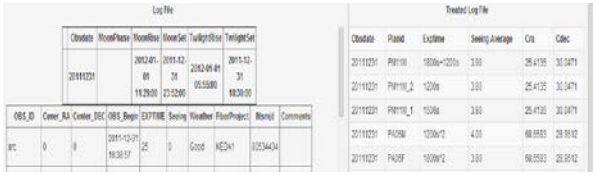

Fig.4. Log File Info

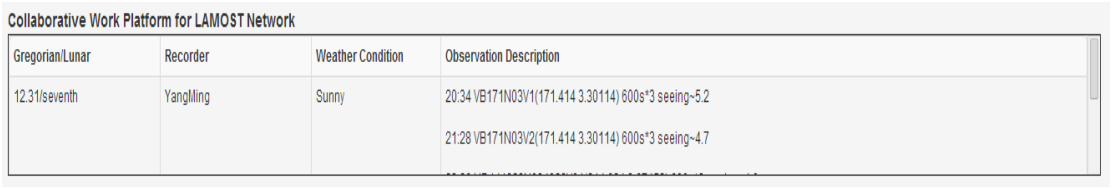

Fig.5. CWPLN Info

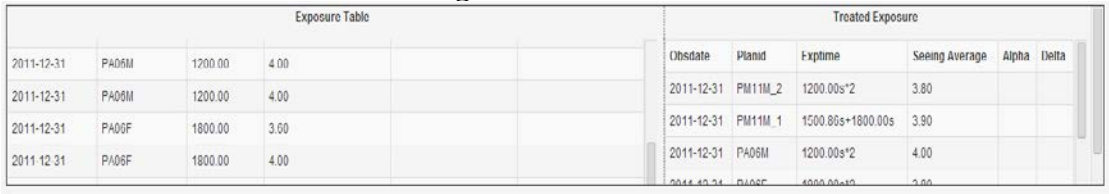

Fig.6. Exposure Info

Fig.7. Sky Info

As shown in Fig.7, the system need to check whether the records of sky information and spectrometer information repeated or not. We use the function “each() ” and “ajax” of JQuery when traversing the table[3][4][5]. The data format we use is “json” and in the background we use “fastjson” to convert the data transmitted from the front.

Logged Process

The page “instore log” lists the dates in reverse order in which the log information has been saved. Similar to unlogged, in this page, there are also two choices for users to select. The page “view or modify the saved log” is very similar the page “fill log information”, so we do not want to repeat it.

Auto-Storage

The data which had been saved can be modified by users in the page “view or modify the saved log”, so it saves the staff’s time. In this page, users need to input the commencing and end date to start the auto-storage. The operation in the background will vary greatly in time performance according to the data range users selected. In the operation, some logs may not be put into database successfully because of the server time out or empty sky number. In that case, the system lists the dates with the two conditions above for users to do manual storage.

Get Log Implementation

The log file of observation is an html file and the system needs to get the file. In this paper, we use httpclient[6] to get the log file.

The content got from the response of httpclient is in the format of “String”. We use Jsoup[7][8] to deal with the content. It can convert the html string to “DOM” (Document Object Mode) and it provides some methods to manipulate the elements in the html file.

Auto-Storage Implementation

The log files are in the website of NAOC. In the operation, the system need to request the log file more than one time. It need to sign in the website before we get the log file. In order to improve the efficiency of request, we log on the website only once with httpclient, then it creates an instance of httpclient, we can use the same instance to request the log file repeatedly.

In the auto-storage, because of many times of request, server time out may happen. In the system, we use “ExecutorService[9]” and “FutureTask” to control the request time. In the first request, 5 seconds can be spend to execute at most. If the first request fails, the system will give the second request 10 seconds, and the third request 20 seconds. If all the three requests fail, the system will list the dates aborted by the program for users to do manual operation.

Acknowledgement

In this paper was sponsored by The Natural Science Foundation of China (Project No. U1431102). *Corresponding author: Jingchang Pan.

References

[1] Jyotishka Datta,Jayanta K. Ghosh. Bootstrap—An exploration[J]. Statistical Methodology,2013.

[2] Wenping Li,Weiqing Wang,Yi Zhao. Research and Development on Survey and Statistical Analysis Software of Resident Travel OD Based on B/S Mode[J]. Procedia - Social and Behavioral Sciences,2013,96

[3] Ali Mesbah,Arie van Deursen. A component- and push-based architectural style for ajax applications[J]. The Journal of Systems & Software,2008,81(12):2194~2209.

[4] Ming Ying,James Miller. Refactoring legacy AJAX applications to improve the efficiency of the data exchange component[J]. The Journal of Systems & Software,2013,86(1).

[5] Cornelia Györödi,Robert Györödi,George Pecherle,Tamas Lorand,Roşu Alin. Web 2.0 Technologies with jQuery and Ajax[J]. Journal of Computer Science and Control Systems,2009,2(2):11.

[6] Pravetz J D, Sites R L. Communicating data using an HTTP client: U.S. Patent 7,272,628[P]. 2007-9-18.

[7] Xu Z, Yan D. Designing and Implementing of the Webpage Information Extracting Model Based on Tags[C]//Intelligence Science and Information Engineering (ISIE), 2011 International Conference on. IEEE, 2011: 273-275.

[8] Gupta S, Kaiser G, Neistadt D, et al. DOM-based content extraction of HTML documents[C]//Proceedings of the 12th international conference on World Wide Web. ACM, 2003: 207-214.

[9] David E. Singh,Florin Isaila,Juan C. Pichel,Jesús Carretero. A collective I/O implementation based on inspector–executor paradigm[J]. The Journal of Supercomputing,2009,47(1):53~75.