

## Design and Implementation of the Scheduling Distribution Module of EDM System

Xiaoqun Liao, Yuanlong Zhang, Shancheng Tang, Anxin Zhao

Institute of Communications Engineering, Xi'an University of Science and Technology  
Xi'an, China

E-mail: zhangshunlong.2008@163.com, 415926304@qq.com, 408291872@qq.com, 27553331@163.com

**Abstract**—Electronic mail play an important role in the enterprise marketing, at present, due to low success rate of send and uncontrollable and so on many kinds of reasons, lead to far cannot satisfy the requirements of user's application. Using method of object-oriented design, integrated multithreading technology, Quartz scheduling framework and ActiveMQ news service framework, design the scheduling module of electronic mail system, realize the communication process, operation process and operation process of distribution. Now the mail system to send mail success rate is 34%, after using the EDM system scheduling distribution module, post success rate as high as 85%. In the laboratory environment, building the mail server and EDM system scheduling distribution module, test for scheduling, mail delivery success rate to meet the design goals.

**Keywords**—Electronic mail ; marketing ; Scheduling ; Object-oriented design

### I. INTRODUCTION

With the rapid development of Internet, email has become an inseparable part of people's life. Email as today's most popular, the fastest and the cheapest way of information transmission, also become a major method of network marketing[1]. Email marketing system have accelerated the promotion of network marketing and a large quantity of small and medium-sized enterprises to use, according to the CNNIC (China Internet network information center) in the first half of 2012 the Chinese small and medium-sized enterprise Internet application survey report shows that the Internet is still the most popular with small and medium-sized enterprises of marketing channel, and the demand for Internet marketing in the development of enterprises is the largest[2]. Traditional email distribution system already cannot satisfy the needs of customers, to design email marketing system and it is necessary to meet the requirements of current customers.

Existing E-mail marketing system high coupling, poor portability, mail delivery success rate low, for mail delivery situation is uncontrollable [3]. This design adopts J2EE technology architecture, with the object-oriented design method, combined with ActiveMQ and Quartz plug-in for scheduling module design and implementation.

### II. THE ELECTRONIC MAIL SYSTEM INTRODUCTION

The email marketing system can be divided into three parts: the Web application layer and scheduling layer and an SMTP server. As shown in the figure below.

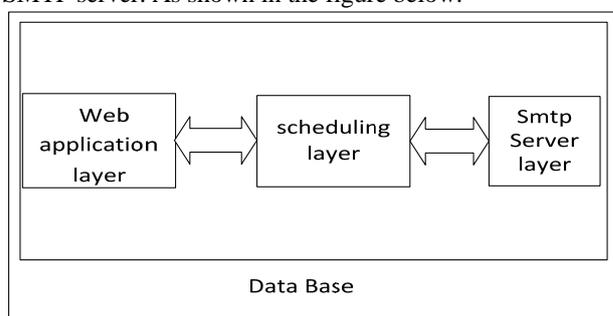


Figure1. Email distribution diagram of the system as a whole

We use Java + struct2 + Exits framework for Web application layer to present a simple, brief, beautiful and easy-operational Web interface to users. Besides, all kinds of manual operations can be done on the Web, such as adding, deleting, modifying, checking E-mail and examining the success rate of sent E-mails [4]. All the data will be send and store in the data base.

The Main Responsibilities of Scheduling layer are processing the user data, dispatching information of data base and finishing the corresponding works [5]. The Scheduling layer configured sending frequency to control the sending rate of assignment. E-mail sender will be called to send E-mails which has already been arranged by a certain arrangement rule.

A SMTP server layer has been designed to send E-mail and return results (success or fail) and reason using Mail server.

JDBC has been used as communication interface between Scheduling layer and Web application layer. The communication between scheduling and Mail server is based on standard SMTP (RFC821) protocol and standard mail format (RFC822) has been used in data structure. Then, we mainly introduce the design and accomplishment of Scheduling layer in this E-mail system.

### III. THE DESIGN AND IMPLEMENTATION OF THE SCHEDULING MODULE

#### A. The design of the scheduling module

In order to meet the practical requirements of users and against low mail-sending success rate as well as uncontrollability, we use method below to design scheduling module. The Scheduling module adopted object-oriented design method and combined multiple frameworks.

1) *The main technique adopted in degree-design:* J2EE is a unique technical architecture which improved application system in transportability, security and recyclability. It also simplified as well as regulated the development and disposition of application system. Because the core technique of J2EE contained many Components, Service Frameworks and technical levels that had common criterion and norm, it got good compatibility on different platforms.

ActiveMQ follow JMS1.1 and J2EE1.4 specifications of the JMS Provider implementation, provides high speed message persistence, support for multiple transfer protocol, etc.

Quartz, entirely written by java, is a Open-source assignment scheduling framework designed for J2SE and J2EE application. Without sacrificing the simplicity, Quartz provides tremendous flexibility and supports database, JavaMail and so on.

2) *Scheduling module design:* Using object-oriented design method, we divide scheduling into five parts: interrogator, task, ActiveMQ, scheduler and mail sender. The figure 2 showed the design of the scheduling module structure.

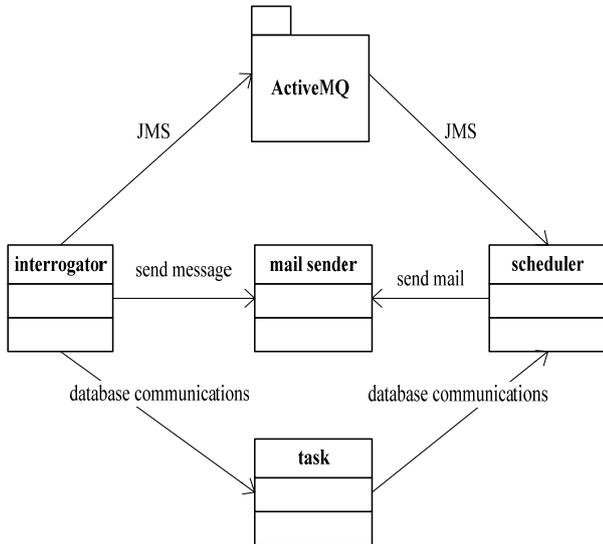


Figure2. Scheduling module design structure

Interrogator: regular query on assignment; return assignment information inquired by DB. At the same time, it send assignment information to Message server

(ActiveMQ) in message way and send a assignment message to mail sever. Then, the mail server created corresponding user name and password according to the previous assignment message.

Assignment-Access database through JDBC; modify assignment in data; provide data service for Interrogator, scheduler, and mail server.

Scheduler-obtain messages from ActiveMQ and decompose as well as arrange assignments by access to DB based on its content, then put assignments into mail sender; Configure parameters of Domain Name in sent mails, which contains data for mail quantity control, results after STMP-log analyzing and reason of sending failure.

Mail sender-generate user name and password according to the message sent by Interrogator; send the scheduling assignments out and return sending results.

Message server: simultaneously monitor three messages, which are mail-server message, assignment message and mail-result message; respond to these three messages and act differently in different parts.

#### B. The realization of the scheduling module

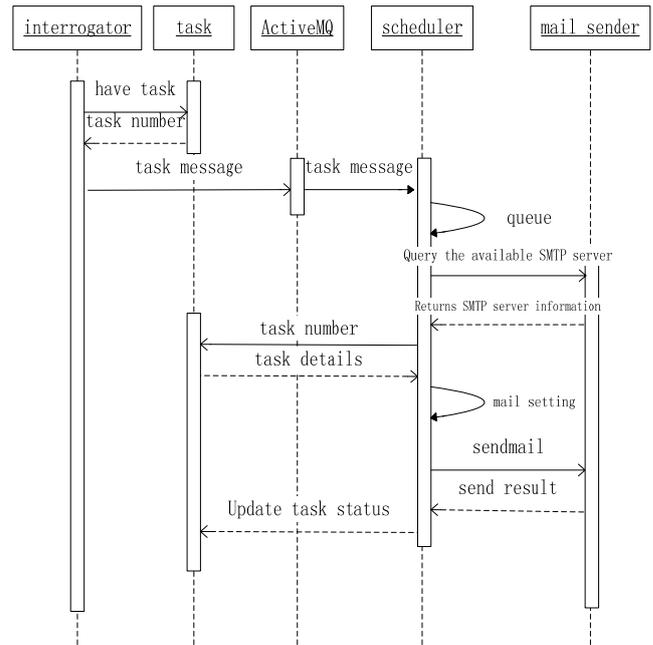


Figure3. Scheduling sequence diagram

1) *Scheduling the operation of the process:* Figure 3 is an overall description of the entire communication process. First, the Interrogator inquires assignment, and then the assignment returns an assignment message back to Interrogator. After got the assignment message, the Interrogator sends a message monitoring by scheduler. When scheduler finds the assignment message, it returns an assignment number to assignment. By analyzing assignment number in DB, the assignment returns analysis

result back to scheduler. Next, scheduler sorts all assignments, configures E-mail and inquires available mail server. If there is any available mail server, scheduler will send the E-mail. The mail sender returns the result information back to scheduler after sending succeed. According to the content of the message, scheduler configures the status of assignment and counts the successfully send number.

2) *The realization of the interrogator module:* The work of Interrogator contains query interval configuration for assignment, assignment inquire implementation and sending assignment message to mail server. First, we need to configured inquire time rule and begin the interrogation. Here, we got a scheduler instance (schedulerFactory). Second, we create a JobDetail instance (jobDetail). By calling assignment inquire method in instance, we can figure out the assignments met the checking condition in Job table and return the results back to InterrogatorJob. If any assignment exists, we call sendMessage method in mail server, which can send message through message creator. Sending succeed, it returns true.

Interrogator main implementation code is as follows:

- a) *Query whether there is task regularly:*

```
interrogator.setTimer(10);
interrogator.start();
idList = mjp.checkNewJob();
```
- b) *The query to the assignments in the form of a message sent to the ActiveMQ:*

```
qser.sendMessage(InterrogatorConfig.getInstance()
.getActiveMqAddress(),InterrogatorConfig.getInstan
ce().getActivemqMsgNameJob(),(Serializable)
idList);
```

3) *The realization of the scheduler:* After scheduling start, we enable the entrance program DispatcherCore, and start monitoring assignment message in ActiveMQ by calling onMessage method. If any assignment message passes, we call JobId to obtain detailed information of parsed assignment. Then, we add this detailed information into DistributePool by calling addMailToPool method then put these information into sending result set with defaulting the unassigned mail domain name. Afterwards, we put objects of different domain name into different threads with instantiating one object(manager) of thread manager and one vector array object(SendMailThread). If the thread number greater than one, we call Wakeup method to wake up other threads and make sure different kind of mail going to their corresponding SendMailThread to send.

4) *Task distribution of the implementation of the process:* Scheduler enters into sending status when it got assignment message like figure 4 showed. The whole sending process will change by the status value. Here's the detailed Description: scheduler configure status value as 1 just after turning into sending status. Then it checks assignments added in the pool, searching for assignment information which status value is 0, 2 and less than 3. These information includes pool ID and email address. If there is no assignment matched, the status value changes to 0 and the thread turns into sleeping status. Otherwise, the status

value changes to 2 and one thread will be enabled. Then, scheduler will try to find a spare domain name to send the assignment by calling lookup method. It also inquires mail server send domain by calling getAvailableServer stored procedure. If there is no matching spare thread, change status value to 4 and make thread change into sleeping status by calling waitfor method. After sleeping for 60 seconds, it resets status value to 1. If any available thread exists, change status value to 3 and send assignment by send method. Sending succeed, it reset status value to 1.

The main job of send method is to send the assignment. If mail successfully sent, it will delete this mail in sending pool and update sendSuccess field in DistributeResult. If failed, it will delete this mail after 3 times repetition or keep retrying until the assignment successfully sent.

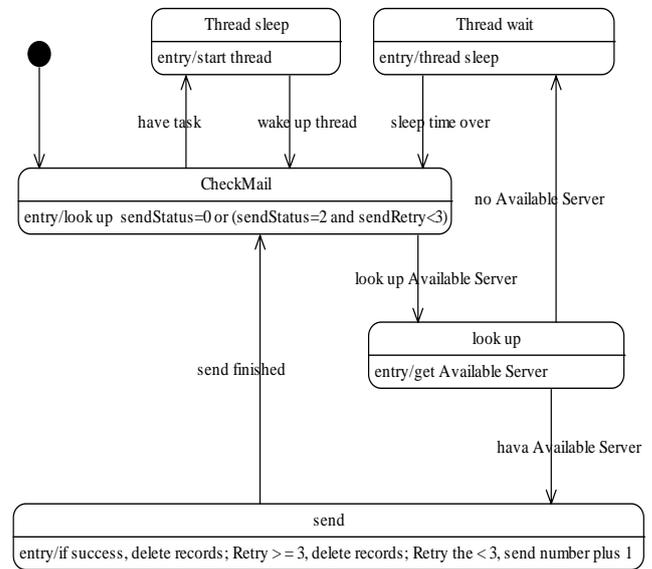


Figure4. Task send state transition diagram

#### IV. SCHEDULING THE TESTS

##### A. The test environment

The test environment-A PC; AN IBM server installed FreeBSD 8.2 OS; Web server installed 2.2.22 Apache with ssl module, DB/DS and postfix-2.9.1 supported.

Set up Web environment; install MySQL5.5.22 database and ActiveMQ message server and JDK. Then put the foreground application on tomcat, the system is running normally.

Set up scheduling daemon, packed the scheduler to be included into the IBM server, run stable.

Collecting 100 valid email addresses, there are 5 different domain names, qq.com, respectively, 163.com, 126.com, yahoo.cn, other different domain name (individual).

### B. The test results

The test results-There is 100 emails with 85 emails sent out successfully, and 15 emails failed for various reasons.

### V. CONCLUSION

This paper mainly discusses the design and implementation of distributed scheduling module in EDM system. Compared with old system, the test result shows the improvement in delivery success rate and reduction in coupling factor of our present system. This design meets users' needs and well adapted in E-mail marketing for SMB. The elevation of delivery success rate gives an important enhancement for enterprise marketing and has a boosting effect to the development of economic in E-age.

The scheduled test analysis shows distributed scheduling module improves mail delivery success rate. From a scheduled test analysis, the use of distributed scheduling module, improve the mail delivery success rate.

### REFERENCES

- [1] Miao Qijun, "Email marketing present situation and development of China", *Journal of economist*, 2004, 12:105-106.
- [2] Qiu Hongmao, Xu Zhaoyang, and Gai Lei. Implementation of the Web Mail System Based on Javamail, *Computer Applications and Software*, 2005,06 : 36-37,111.
- [3] Shi Zhiguo, *The JSP application tutorial*, Beijing: The northern jiaotong university press, 2008, 56-60.
- [4] Yuan Meileng and Nie Zhe, Use JavaMail/JAF development Chinese mail system key technology research, *Computer applications and software*, 2008,03.
- [5] Xu, Youming and Li Chenhui, "Based on the.net platform email delivery system design and implementation", *Journal of guilin aviation industry college*, 2009,02.
- [6] Song Wei, Shen Jianxiong, and Sun Yi, "Large mail server performance testing methods and practice", *Computer applications and software*, 2010, 12.
- [7] Ni Minwei, Pan Yinrong, and Hu Youhua, "Heart beat monitor design and implementation based on JMS asynchronous messaging". *Computer applications and software*, in April 2007, 24 (4) : 96-98.
- [8] Yin Bin, Jiang Chongli, and Dong Ming, "Web mail system based on Java", *Computer engineering*, July 2004, 30 (14) : 183-184.
- [9] Guo Guangjun, Hu Yuping, and Dai Jingguo, "Based on the Java multi-thread parallel computing technology research and application", *Journal of central China normal university (natural science edition)*, June 2005, 39 (2) :169-173.
- [10] Lin Panfeng, Wang Hanhu, Zhang Ruliang, etc. "A new email system in the research and simulation", *Computer applications and software*, 08, 2009:49-51.