An Implementation of Embedded Geographic Information System Based on Cloud Computing

LU Yiqin, YU Kanghua, LIU Yuan
School of Electronic and Information Engineering,
South China University of Technology, Guangzhou, China
liuzq715@163.com

***Abstract.* An implementation of embedded geographic information system is proposed based on cloud computing mode. With cloud computing, the newest geographic information can be gotten in the terminal, meanwhile, good mobility and real-time performance can be obtained. MapX is used in cloud server to accomplish the integrated developing of GIS server. The geographic information is located there, so that it is convenient to be updated and maintained. J2ME technology is employed in terminal to accomplish GIS presentation in embedded platform. Socket is used to realize the communication between terminal and the server. With cloud computing, the running effect in terminal can be ensured, for the complicated processing of the map data is done in cloud server instead in the embedded terminal.**

Keywor*ds: cloud computing, GIS, embedded system, J2ME, MapInfo*

I. INTRODUCTION

Cloud computing, raised by Ramnath K. Challappa in 1997, is now regarded as a new-generation computing mode. In 2001, Eric Schmidt, the CEO of Google firstly proposed the concept of cloud computing on the *Search Engine Conference*. In 2004, Amazon promoted the cloud computing service and became one of the few cloud computer providers which guarantee the 99.95% normal operation time. In addition, Microsoft, Salesforce, IBM, EMC, NetApp and Apache Hadoop etc, promoted their own cloud computing platforms [1]. According to the *Top 10 Strategic Technologies* identified by Gartner, cloud computer had been selected from 2009 to 2011, ranking No. 1 during the past two years [2].

This paper proposes an implementation of embedded GIS (Geographic Information System) based on cloud computing mode. With this mode, the newest geographic information can be gotten in the terminal. Meanwhile, good mobility and real-time performance can be obtained. The geographic information can be updated and maintained in the server.

GIS has high requirements for computing, storage and reliability, while the service provided by cloud computing can meet the GIS requirements, for example, *ArcGIS Online* of ESRI and *GoogleMap* of Google are not only the frameworks of cloud computing, but also provide GIS service by a series of API [3, 4]. Zhou, et al pointed out, the main characteristics of GIS software engineering under the cloud computing environment lied in that, the basic datum and applications are provided by specialized institution as the basis for the structuring more complicated GIS application system. From the respect of GIS software engineering, he analyzed the influence of cloud computing on GIS software development, and discussed GIS software engineering design based on cloud computing, with respect to software system structure, developing organization and management [5]. Zhao, et al proposed a GIS model based on cloud computing, namely *GISCloud*. The cloud computing defined in this model is as followings: multiple servers are connected through internet and form a *GIS Cloud*; GIS application is provided as the basic service. Deployment and configuration of the cloud resources are made dynamically according to GIS application requirements. Application services are provided to the upper level by means of virtualization of the cloud resources [6].

With respect to the application of the embedded GIS, the geographical information in the positioning devices in the market mostly is stored in the hard disks or CDs of the navigators, PDAs or of notebook computer. This will result in inconvenience to the information update of the users, and will not facilitate the uniform management. In this paper, an embedded GIS application system based on cloud computing is proposed. With cloud computing, electronic map information and GIS servers both run on the remote servers. In such a system, the geographic information is easy to update. During the update of the geographic information, it is not required to change the terminal, but to change the geographic information in the server, which facilitates the unified management and update of the geographic information. Such design has another advantage that it satisfies the characteristic of embedded system. Since the embedded device has the limited capacity of computing and storage, it is required to reduce the data processing at the terminal to ensure the processing efficiency at the terminal. The terminal in this design sends the instructions and initial data to the servers, mainly by wireless network, so that data processing is completed in the servers. As the results, the grid map in `the PNG format is sent back to the terminal. Therefore, the complicated data processing and analysis on the embedded device are avoided, operation efficiency at the terminal is ensured.

II. COMMUNICATION MODE OF THE SYSTEM

In this paper, C/S mode is adopted for the embedded terminal and server, which is different from that for current WebGIS system. In a common WebGIS, the popular B/S mode is often employed. Since web browser is used at the WebGIS terminal for processing and display, there may exist two problems during the use in the embedded system: first, if the vector map is used in the user terminal, it is required to install a plug-in at the terminal, or automatically download

Java Applet or ActiveX control during running, but not all the browser in the embedded devices support the installation of such ActiveX; secondly, in case that the grid map is used in the user terminal, it is not required to install additional plug-in program, but it is difficult to finish the data acquisition and processing tasks, such as obtaining the information of longitude and latitude, data encryption and decryption, etc. Though there is wide universality at the WebGIS terminal, it is uneasy to realize expansion and customization of personalized functionality. This is the why the basic C/S mode is adopted instead of B/S used in WebGIS in this paper.

The socket mechanism [7] is used for communication between the servers and the terminals. There are two kinds of socket: *SOCK_STREAM* and *SOCK_DGRAM*. SOCK_STREAM is a connection-oriented socket, mainly for TCP applications, while SOCK_DGRAM is connectionless socket, mainly for UDP applications. To ensure the reliability of the network communication, SOCK_STREAM is employed in this paper. For the socket built on this type, the data can be bidirectional stream of bytes without the limitation of length.

III. IMPLEMENTATION IN THE SERVER CLOUD

The server part of this system is the key to data processing and map positioning. In a cloud computing mode, a virtual server for the system must be provided by the server cloud. Map X control is employed in the server part. MapX organizes and process the map datum in a layered-mode, i.e., a map is processed into multiple overlapping transparent layers. Every layer includes a different method of the whole map. For example, the first layer is the *regional layer* including the boundary, the second layer is the *linear layer* including highway and the third layer is the *spotted layer* including the cities. The three layers form a complete and useful map by overlapping one by one.

The map used in this paper includes the frames, subway lines, the names of subway stations, ground features, elevated roads, surface level 2 of the elevated roads, surface level 3 of elevated roads, the paths in parks, lakes and reservoirs, residential lands, overpasses, the sidelines of overpasses, floor 1 of the overpasses, floor 2 of the overpasses, floor 3 of the overpasses, floor 4 of the overpasses, the names of the overpasses, green, county border, names of mountains, downtown class-1 roads, the surface layers of downtown class-1 roads, downtown class-2 roads, the surface layers of the downtown class-2 roads, downtown class-3 roads, the surface layers of the downtown class-3 roads, downtown class-4 roads, the surface layers of the downtown class-4 roads, double-line river, railway and so on.

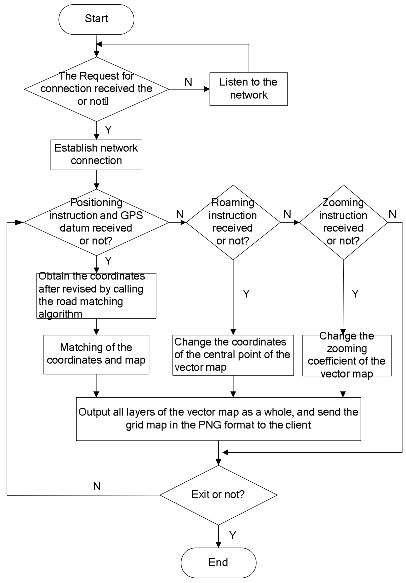

Figure 1. The flow chart of server part of the system

Fig. 1 shows the flow chart of the implementation of the server part. Firstly, the server listen to the network to see whether there is a request for network connection from a client in the user terminal. Once the request is received, the server establishes network connection and wait for further command from the client. If a request for positioning from the client is received, the server obtain the initial GPS positioning datum from the received information, then the road matching algorithm will be used for the revising the data of longitude and latitude, finally, the positioning point on the vector map in MapInfo format is located.

If there is request of map roaming sent from the client, the server will modify the coordinates at the central point of the vector map and refresh the map. If the client sends a request for map zooming, the server will modify the zooming coefficient of the vector map. After the processing of the requests from the client, the server will take all layers of the vector map as a whole and output it as the grid map in PNG format, and then it will be returned to the client [8-9].

IV. IMPLEMENTATION OF THE CLIENT IN THE USER TERMINAL

The functionalities of the client of this system mainly consist of two parts, one is *positioning*, another is *map roaming*. When the positioning is select by the client at a user terminal, GPS datum is read from the GPS receiving module, which are in the NMEA-0183 format. After simple processing of the GPS datum, socket connection to the servers is established, and the client will send the datum and positioning instructions to GIS virtual server. After the data processing in the server, such as road matching, the server will find out the positioning point on the vector map in the MapInfo format, then output the grid map in the PNG format, and send the map to the client after encryption processing. When receiving the map, the client will display the map after the decryption processing.

When the map roaming and zooming function is selected by the client at a user terminal, the client should firstly establish the socket connection with the server, then send the roaming and zooming instruction to GIS server. After finish the processing of map displace and zooming, the server will output the grid map in the PNG format, and send it to the client after encryption. After receiving the map, the client displays the map after decryption.

J2ME technology is used for the client application programming. Since the user terminal is the embedded device, which has the limited processing capacity and the limited storage space, the data processing and map transformation will be finished in the server to ensure the high-efficiency of the client.

The GPS data format of NMEA-0183 is the standard format developed by National Marine Electronics Association (NMEA) for marine electronic devices. In the present, it has become the uniform standard protocol of GPS devices of RTCM (Radio Technical Commission for Maritime Services), and almost all GPS receivers support this format [10].

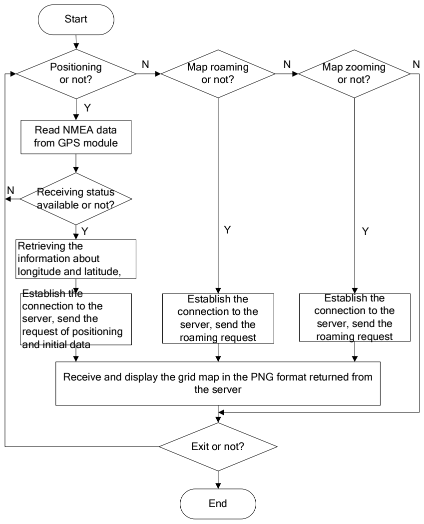

Figure 2. The flow chart of client part of the system

The flow chart at the client is shown as Fig. 2. When the positioning function is chosen at the client, the GPS data will be read from the GPS module in the NMEA format. After retrieving the information about longitude and latitude, the client will establish the socket connection to the server, and send the GPS datum and positioning instruction to the GIS server.

When the map roaming and zooming function is chosen at the client, the client will firstly establish the socket connection with the server, and send the roaming or zooming instruction to GIS server. After the processing of relevant instruction and datum, the server will output the grid map in the PNG format, which may be displayed by the client after received and by the terminal. Encryption will be used during the transmission [8-9].

The applications of the client are developed based on WTK platform, which is the wireless developing tool provided by Sun. WTK can be integrated into the popular IDEs (Integrated Development Environment), such as JBuilder and Eclipse, for improving the development efficiency.

To develop a MIDP application programs on WTK programming, the class of *javax.microedition.midlet.MIDlet* shall be inherited. The class of *GISClient* designed in this paper is inherited from the class and abstract three methods, i.e. *startApp (), pauseApp() and destroyApp()*. The method of *startApp()* is generally used to start or restart a MIDlet, which may be invoked by the system under any situation, aiming at the request for resources required by MIDlet. The method of *pauseApp()* is used by the system to request pause of MIDlet. With the combined use of the methods of *pauseApp()* and *startApp()*, the resources can be released as many as possible. The method of *destroyApp()* is used by the system in the case that the class of MIDlet is going to be eliminated. It can also be called directly by the class of MIDle itself. The two methods are designed for the necessary cleanup before the withdrawal of the MIDlet class, so that the occupied resources can be released. Because they are abstract methods, they should be *implemented* in the specific MIDlet.

The J2ME applications of the client consist of seven classes. In the main class of *GISClient*, an object of *MainModule* is instantiated, which is inherited from the class of *Canvas*. *Canvas* is the parent class of all lower user components. The method of *paint ()* of the *Canvas* is implemented to draw on the screen. By calling the method of *display.setCurrent()* of the main class, the map canvas can be displayed on the screen of a mobile phone.

An object *SerialReader* is created as an instance of the class *MainModule*, which is responsible for reading the positioning information from the serial port connecting to GPS module; and retrieving the datum of longitude and latitude with *Parse*. The datum is packaged into *Record* and stored in record buffer. Then *MainModule* retrieves a *Record* from the record buffer, and sent it to the server through socket; meanwhile, it receives the map datum returned from the server, and displays them on the screen. To ensure the above processes runs in a parallel way, the multi-threading technology is used.

V. THE TESTING RESULTS

The applications of the server part of this system runs in a virtual server provided by a PC server group, and the client runs on an ARM9 embedded system based on the Samsung S3C2410X processor. Arm-Linux embedded operating system with open source codes developed by MIZI Company in South Korea is installed. J2ME operation platform is built by migrating KVM and MIDP, provided by SUN. The user terminal is connected to the server through an IP network. The running at the user terminal is shown as in Fig. 3 [8-9].


Figure 3. The running results of the client in a embedded system

The client in Fig.3 use road matching technology for positioning. The black “x” in the map represents the user in movement. The client sends to the server the information about longitude and latitude, meanwhile it receives and display the map returned from the server. The map is in the PNG format. It can be seen from the map that because of the road matching algorithm, the user’s position matches well the third-level roads around East Lake.

VI. CONCLUSIONS

In this paper, an embedded GIS application based on cloud computing has been designed. The system has the functionalities of GPS positioning, map roaming and zooming. The client of the system is realized with J2ME technology. J2ME has been simplified on the basis of Java standard version, however, the class libraries supporting the network programming and serial programming are maintained. These class libraries ensure the realization of the client programs. Therefore, by migrating the J2ME operation platform to the embedded system, the client can runs in an embedded device. The server part of the system runs on a virtual server provided by the server groups. Some controls are employed for developing of the server part, such as MapX.

The system is based on cloud computing mode, the complicated data processing and analysis runs on the virtual server, which ensures the operation efficiency of the embedded terminal. Besides, geographic information is maintained and updated in the service in a centralized way, the terminal can obtain the newest geographic information without maintenance.

ACKNOWLEDGMENTS

This work was supported by National Science & Technology Pillar Program(2008BAH37B05011), Guangdong Foundation of Science and Technology Project (2008B090500073, 2010B090400261).

REFERENCES

[1] A. T. Velte, T.J. Velte and R. Elsenpeter, Cloud Computing: A Practical Approach, The McGraw-Hill Companies, (2010).

[2] Gartner, Gartner Identifies the Top 10 Strategic Technologies for 2011, http://www.gartner.com/it/page.jsp?id=1454221

[3] Y. Liu, G. Lei. The development trend of the GIS software technology (in Chinese), China's Foreign Trade, No.12 (2010), p.197

[4] L.Q. Dai, N. Chen. The development of GIS in the times of cloud computing (in Chinese), Journal of Anhui Agricultural Sciences, Vol.37, No.31 (2009), p.15556

[5] P. Zhou, F. Yin. GIS software engineering model based on cloud computing technology (in Chinese), Bulletin of Surveying and Mapping, No.11 (2010), p.22

[6] W. Zhao, Q. Geng. Application of cloud computing to GIS model (in Chinese), Geospatial Information, Vol.8, No.6 (2010), p.8

[7] X.F. Hu, C.G. Zhang,. Apply of socket technology on embedded gateway and development of Java (in Chinese), Industrial Control Computer, Vol.19, No.2 (2006), p.21

[8] K.H. Yu. The Research and implement of Embedded GIS based on J2ME. Master Dissertatio, South China University of Technology (2006)

[9] Y. Liu.. A Kind of J2ME-based Embedded Secure Terminal for Military Communications. Master Dissertatio, South China University of Technology (2010)

[10] T.T. Cao, Y. Gao. Application of NMEA-0183 protocol for GPS，Electronic Engineer, Vol.32, No.10 (2006), p.8

\