Pseudo-Satellite Data for Monitoring Cloud Platform

Guofu Wang*, Zhen Li², Jincai Ye, Faquan Zhang, San Huang
Guilin University of Electronic Technology, China

*Corresponding author e-mail: guofwang@126.com, *524691955@qq.com

Abstract. This paper proposed a scheme for monitoring using pseudo-satellite networking aiming at the landslide phenomenon in China, and to make up for the shortcomings of traditional testing methods, this paper used cloud computing related technologies to realize back-end data monitoring and early warning services. The front end developed the Web side and the Android side for users to use. This cloud platform has strong communication performance and data processing capabilities, its working status as a whole meets the requirements.

1. Introduction
Landslides in China are always an important research topic. Due to its strong anti-interference ability, flexible networking and good economy, pseudo-satellite is widely used in special scenes such as mining areas, tunnels and high-rise “urban canyons” in the valley. At the same time, certain technical means are needed to conduct efficient and rapid real-time monitoring of the pseudo-satellite monitoring area. Cooperating with cloud platform to build corresponding data processing and management system is of great significance for real-time monitoring.

2. Cloud platform overall architecture
The overall architecture of the cloud platform is divided into three parts. The first part is the front-end monitoring service, the second part is the back-end data monitoring and early warning service, and the third part is the monitoring node based on the pseudo-satellite networking system. The overall architecture is shown in Figure 1.

![Cloud platform overall architecture](image-url)
3. Cloud platform overall architecture

3.1. Fundamental

Pseudo satellite network system shown in Figure 2, the monitoring node receives signals transmitted by 4 pseudolites. In an ideal situation, the distance between the pseudo-satellite and the receiver can be obtained by obtaining the difference \( t_u - t^s \) between the time \( t_u \) of the receiver receiving signal and the time \( t^s \) of the pseudo-satellite transmitting signal. Its formula is:

\[
C \times (t_u - t^s)
\]

Where \( C \) is the speed of light.

![Pseudo-satellite network monitoring diagram](image)

**Figure 2.** Pseudo-satellite network monitoring diagram

However, the measured signal propagation time in the actual test will have the clock synchronization between the pseudolite and the receiver and the influence of environmental factors. Considering the clock error \( \delta t \) and the delay caused by the environment I can be obtained as follows:

\[
t_u - t^s = \tau + \delta t + I
\]

Where \( \tau \) is the true propagation time of the signal. Let \( \rho_c = t_u - t^s - I \), then the above formula can be rewritten as:

\[
\rho_c = \tau + \delta t
\]

Multiply both sides of the formula by the speed of light \( C \), available:

\[
C \cdot \rho_c = C \cdot \tau + C \cdot \delta t
\]

Where \( C \cdot \tau \) is the true distance between the pseudolite and the receiver, which is \( \sqrt{(x_1 - x)^2 + (y_1 - y)^2 + (z_1 - z)^2} \), where \( (x, y, z) \) represents the unknown coordinates of the receiver position, \( (x_1, y_1, z_1) \) represents the known coordinates of pseudolite 1. Combined with 4 pseudosatellites, a quaternary nonlinear equation system can be obtained:

\[
\begin{align*}
\sqrt{(x_1 - x)^2 + (y_1 - y)^2 + (z_1 - z)^2} + C \cdot \delta t &= C \cdot \rho_c^{(1)} \\
\sqrt{(x_2 - x)^2 + (y_2 - y)^2 + (z_2 - z)^2} + C \cdot \delta t &= C \cdot \rho_c^{(2)} \\
\sqrt{(x_3 - x)^2 + (y_3 - y)^2 + (z_3 - z)^2} + C \cdot \delta t &= C \cdot \rho_c^{(3)} \\
\sqrt{(x_4 - x)^2 + (y_4 - y)^2 + (z_4 - z)^2} + C \cdot \delta t &= C \cdot \rho_c^{(4)}
\end{align*}
\]

Where \( \rho_c^{(n)} \) can be measured by the receiver, \( \delta t \) is the unknown clock difference. Solving the quaternary nonlinear equations, the position information of the monitoring nodes can be obtained.
3.2. Pseudo-satellite signal navigation message design

The pseudolite's transmit signal consists of three parts: carrier, pseudorandom code, data code. The carrier frequency is 1575.42MHz, pseudo-random code uses pseudo-random code (Gold code) in Beidou navigation system as pseudo-random code [1], and the circuit logic diagram is shown in Figure 3.

![Figure 3. Five-level gold code generator logic diagram](image)

Pseudo-satellite navigation message format reference Beidou navigation message format. Since the pseudolite is fixed on the ground and the number of pseudolites in the pseudo satellite networking system is generally four, the navigation message data format of the pseudolite is different from that of the Beidou navigation message. For example, it does not need to provide satellite orbit parameters, delay correction of the ionosphere, ephemeris of all satellites, etc. Therefore, it is necessary to simplify the navigation message format of Beidou and apply it to the pseudolite system. The simplified navigation message is shown in Figure 4.

![Figure 4. Pseudo satellite navigation message frame format](image)

4. Back-end data monitoring and early warning service design

4.1. Overall architecture design

The backend data in this article uses Java [2], Netty, RESTful API, Spring Framework, Spring Data, Spring Boot [3], MongoDB database, Alibaba Cloud Computing Service, Alibaba Cloud CDN [5].

The development of the back-end data monitoring service can be completed through the appeal base component, framework and third-party services. The service needs to undertake the following tasks:

1. The data collected by each monitoring node is separately summarized and saved in the database.
2. The processed data of each monitoring node is processed by an algorithm to calculate the pseudorange between the pseudolites and the four pseudolites.
3. Analyze the calculated pseudorange, judge that there is an abnormality, and alarm the abnormal data.
4. Provide RESTful API to facilitate the web-side monitoring service, Android-side monitoring service, etc. to obtain various data provided by the back-end service.
According to the above objectives and requirements, the design architecture block diagram is shown in Figure 5.

**Figure 5.** Back-end data monitoring and early warning service overall architecture

4.2. Work process

(1) Data reception

After establishing a TCP connection, the Netty-based TCP service module continuously receives data information from the monitoring node.

(2) Data processing

After the monitoring node completes a round of data collection tasks, the back-end service has completely received all the collected data in this round and has been completely persisted to the MongoDB database.

(3) Storage of processed data

After the pseudo-range and other information processing is completed, the cloud platform calls the data cache and the persistence module to complete the persistence operation of the above data.

(4) Data analysis and abnormal alarm

After the storage operation of the settled data is completed, the cloud platform randomly starts the analysis of the data. When the difference between the two is greater than the set threshold, it will retrieve the web interface module to actively push the abnormal information to the opened web service and the Android service that has started the push function.

(5) Distribution of processed data

When the web service or the Android service is enabled, the two client services will continuously retrieve the backend service and monitor the running status of the node through the Spring MVC-based web interface.

5. Front end design

5.1. Web-site design

(1) Login interface

The background video of the web page is looped as shown in Figure 6. Click the "Login" button, the web application will authenticate the rights through the backend service.
(2) Current status of each node

Expand the “Address Status Monitoring” group in the left navigation bar and click the “Status Overview” option to enter the geological status overview page of each monitoring node, as shown in Figure 7.

5.2. Android design

Since the coordinate information of each monitoring node has been processed by the data and sent to the backend service, the application can conveniently obtain the coordinate information of each monitoring node and mark it in the map, as shown in Figure 8. If the result of the latest task execution of a monitoring node is abnormal, the icon of the node in the map is blue. If the result is abnormal, the icon is red, as shown in Figure 9.
6. **Conclusion**

In this paper, the pseudo-satellite ranging method is used to monitor the landslide, and the cloud computing related technology is designed and implemented to implement the data monitoring cloud platform. The reliability, communication performance and data processing performance of the cloud platform are high, and the overall requirements meet the design requirements and meet the actual needs.

**References**

[1] Simulation on C/A codes and analysis of GPS/pseudolite signals acquisition [J]. Science in China (Series E: Technological Sciences), 2009, 52 (05): 1459 - 1462.

[2] Horstmann C S. Core Java Volume I--Fundamentals [M]. tenth. ed; Prentice Hall Press, 2016, 1-15.

[3] Walls C. Spring Boot in action [M]. Manning Publications Co., 2016, 1 - 22.

[4] Banker K, Bukkum P, Verch Set al. MongoDB in Action: Covers MongoDB version 3.0 [M]. Manning Publications Co., 2016, 3 - 23.

[5] Peng G. CDN: Content Distribution Network [J]. Research Proficiency Exam Report. 2004: 1 - 6.