ISSN: 2710-0588

Remote Serial Port Virtualization Management System based on Brotli Compression Algorithm

Longshan Wu[1], Chunlong Yao[1,*], Ying Qu[2]

[1] Dalian Polytechnic University, Information Institute, Dalian, China

[2] Ocean University of China, School of Economics, Qingdao, China

[*]Corresponding author Email: yaocl@dlpu.edu.cn

Abstract

With the rapid development of information technology, data communication and management systems are becoming more and more important in various industries. Especially in remote device management and data center operations, effective data processing and transmission mechanisms are crucial to ensuring system performance and reliability. In this context, serial port virtualization technology is particularly important because it allows remote devices to be controlled and monitored over the network, so that physical location constraints are no longer an obstacle to efficiency and effectiveness. However, as the amount of data increases, the efficiency of data transmission and storage becomes a key issue. Although traditional compression algorithms solve this problem to a certain extent, they often fail to provide ideal compression ratios and processing speeds when processing large amounts of data, especially in environments with limited bandwidth or limited computing resources. Therefore, finding an efficient compression algorithm has become the key to improving the performance of the remote serial port virtualization management system. This article adopts the method of integrating the Brotli algorithm into the system to solve the data compression and transmission efficiency problems faced in the traditional serial port virtualization system, thereby improving the overall performance and user experience.

Keywords

Brotli; Serial Device; Compression Algorithm.

1. Introduction

Currently, serial port virtualization technology has been widely used in remote device management, Internet of Things (IoT) and industrial control systems. This technology allows users to remotely access and control devices connected to the serial port through the network, greatly improving operational flexibility and efficiency. However, as the number of connected devices increases and the amount of data expands, data transmission and processing efficiency becomes a challenge[3]. In the field of data compression, traditional algorithms such as ZIP and gzip have always dominated. These algorithms perform well in terms of compression ratio and versatility, but their performance and efficiency are often limited in scenarios where large-scale data processing or high real-time requirements are required. In recent years, the Brotli algorithm has received more and more attention as an efficient compression technology. It has demonstrated excellent performance in many fields, especially in network transmission and storage. Compared with traditional algorithms, Brotli provides a higher compression ratio and faster compression speed, especially in text- and data-intensive applications. Although the Brotli algorithm has been widely used in web browsers and mobile applications, there are relatively few studies on its application in remote serial port virtualization management

ISSN: 2710-0588

systems. These systems have extremely high requirements on the efficiency and reliability of data transmission, and the potential advantages of the Brotli algorithm are expected to play an important role in these scenarios[5]. Currently, research on the performance evaluation and optimization of the Brotli algorithm in different scenarios is gradually increasing. Scholars are committed to exploring how to adjust and optimize the Brotli algorithm to adapt to different types of data and application scenarios, but specific application research in the field of remote serial port virtualization is insufficient.

2. Main Technique

2.1. System Architecture and Deployment

As shown in the Fig. 1, the remote serial port virtualization management system based on the Brotli compression algorithm is mainly divided into three parts, namely the PC software used by the PC to connect the serial port device, the server that communicates with the PC for data, and the user. Client for data display and communication. The main job of the PC-side software is to wait for the serial port device to collect data, obtain all the collected data through serialport serial communication technology, and then send it to the cloud server after processing. The cloud server is written in nodejs and relies on Nextjs. The framework is built, and then the data is stored in the Mysql database, and finally the data is sent back to the client website built by the Vue framework, thereby realizing the overall integration of the remote serial port virtualization management system based on the Brotli compression algorithm.

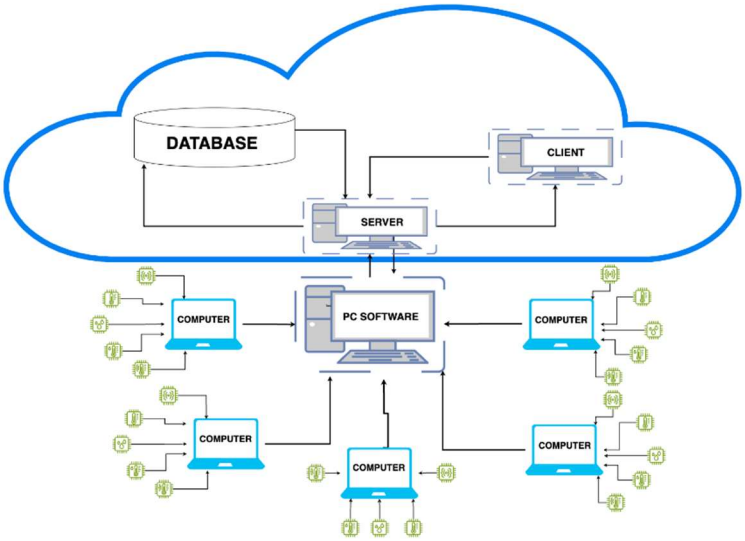

Fig. 1 System architecture diagram

2.2. Brotli Algorithm

As an emerging compression technology, Brotli has attracted attention due to its excellent compression efficiency and speed . It was developed by Google to improve data transfer by optimizing compression speed and efficiency. The Brotli algorithm not only provides a higher compression rate compared to traditional algorithms, but also reduces CPU usage and energy consumption while maintaining compression quality. This makes the Brotli algorithm particularly suitable for scenarios that require efficient compression and fast data transmission, such as remote serial port virtualization management systems[10]. In the field of serial device communication, the Brotli compression algorithm has a series of advantages compared with Deflate, LZ4 and Zstd algorithms. Brotli provides a higher compression ratio than Deflate, which enables it to transmit less data in bandwidth-limited serial communication, thereby improving

ISSN: 2710-0588

transmission efficiency[6]. Compared with LZ4, although LZ4 performs well in terms of compression and decompression speed, Brotli's advantage is that it provides a higher compression rate, which is especially important in situations where large amounts of data need to be transmitted. In addition, although Zstd provides similar compression ratios to Brotli and allows a wider range of compression level adjustments, Brotli performs particularly well for compressing text data and remains efficient in decompression speed, which is important for serial ports that process text instructions and control messages.[8] It is extremely beneficial for the equipment. In summary, Brotli can provide high compression rates while maintaining reasonable decompression speeds, making it an attractive option in serial communication systems with limited resources and the need for efficient data transmission. Fig. 2 and Fig. 3 simultaneously test the compression volume of serial port device communication data of the same size. It can be clearly seen that the Brotli algorithm is significantly better than the other three algorithms in this aspect. Only zstd is not far behind it. , therefore, both compression algorithms can be used as alternatives for remote serial port virtualization management systems.

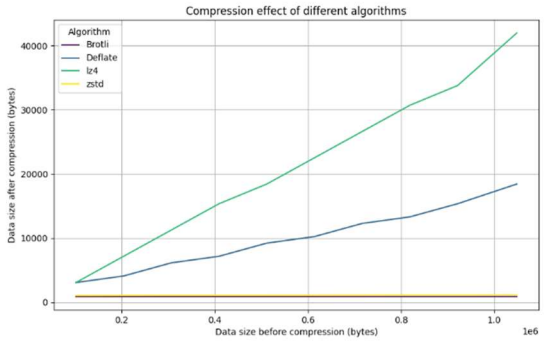

Fig. 2 Comparison of 4 algorithms

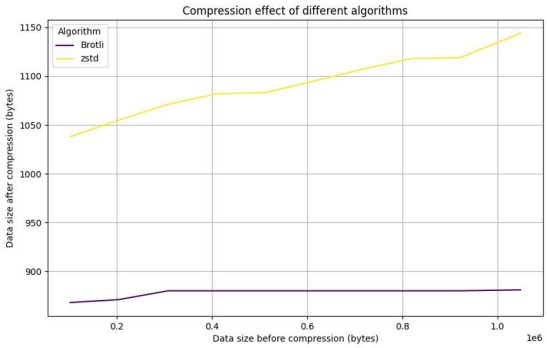

Fig. 3 Comparison of 2 algorithms

2.3. Database Connection

How to connect the system is the first problem that must be solved when the server communicates with the database. The remote serial port virtualization management system based on Brotli compression algorithm uses Prisma technology to connect to the Mysql database. Prisma is an open source, next-generation ORM (Object Relational Mapper) for Node.js and TypeScript applications designed to improve development efficiency and security of database operations[9]. Prisma's main advantage is its simple and intuitive query language, which not only allows developers to perform database queries in a strongly typed manner, but also provides automatically generated and intelligent code completion capabilities. This reduces runtime errors and increases development speed. Prisma supports mainstream database systems such as PostgreSQL, MySQL, SQLite and SQL Server, and provides a powerful, automated set of CRUD operations through the Prisma Client that are easy to understand and use. In addition, the Prisma Migrate function supports database version control and migration, providing convenience for the evolution of database architecture. These features make Prisma an attractive choice for data layer development in modern applications, especially in projects that pursue type safety and development efficiency.

3. System Module Design

As shown in the Fig .4, the system is mainly divided into three different user roles, namely User role, Administrator role, and User administrator role. The business modules of the entire system include: Device management module, Log management module, Rights management

ISSN: 2710-0588

module, User management module, System configuration module, Instruction management module, Monitoring and management module, Data audit module, Data reporting module, Alert configuration module, Early warning notification module, Data large screen module.

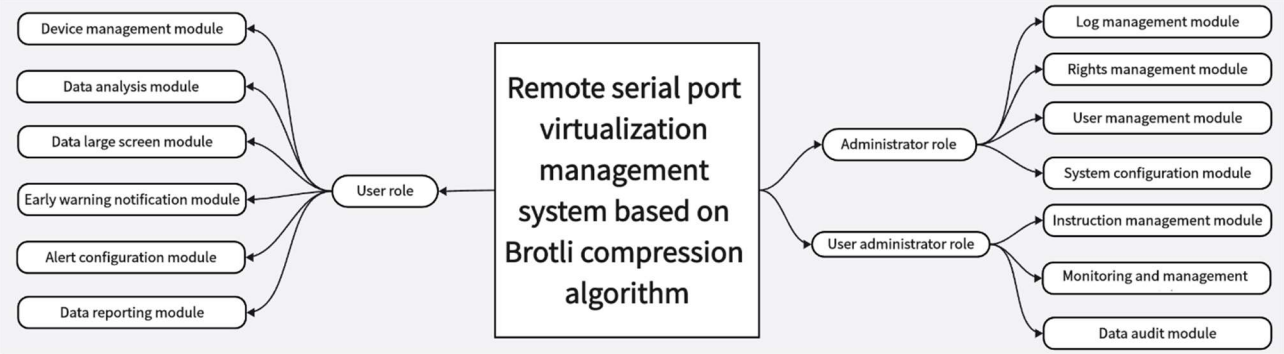

Fig. 4 System module design

3.1. Feature Design

Administrator role is the user with the highest authority in the system. It includes all business function modules of User administrator role and User role. In addition, it also has Log management module, Rights management module, User management module, and System configuration module. Log management module can Check the system log at any time. If there is erroneous data or incorrect operation in the system, you can use this module to locate and deal with the problem in time; the Rights management module can assign different menu permissions and button permissions to different roles, and the User management module can handle the validation of personnel accounts. If it fails, the System configuration module can perform daily analysis of the system's traffic to facilitate operation and maintenance personnel to handle corresponding work.

User administrator role is a high-level user. The difference between them and ordinary users is that they can further review the data reported by users, which is equivalent to the superiors of employees. In addition, they can also centralize the instructions of serial devices in the Instruction management module. management.

User role is the most basic authority role with the largest number of users. In the Device management module, users can freely add and delete serial devices to operate the serial devices. After the connection to the serial devices is completed, the user can add and delete serial devices in the Data analysis module. View the data collected by the device, and all devices under the permissions of this role can be centrally viewed in the Data large screen module . In the Alert configuration module, users can customize the way to configure device alarms, such as alarming when the temperature exceeds 100℃, alarming when the power is below 20%, etc. After setting the corresponding alarm rules, the user can set the alarm in Early warning notification The module checks the alarm information of alarm equipment and processes these alarm information.

3.2. Database Design

The remote serial port virtualization management system based on Brotli compression algorithm uses Mysql database for storage . MySQL is a widely used open source relational database management system, known for its high performance, reliability and ease of use, and supports multiple operating systems. In the serial device virtualization system, MySQL can be used as a powerful back-end data storage solution, providing persistent storage of device configuration, user data, operation logs and other information[7]. Since serial device data usually involves a large amount of time series data, MySQL's advantage is that it can effectively handle large-scale data insertion and query operations, ensuring data consistency and integrity.

ISSN: 2710-0588

At the same time, MySQL's high scalability and mature cluster support ensure that the virtualization system can handle the needs of concurrent access from multiple devices and high data availability, which is crucial to ensuring the stability and response speed of the serial port virtualization system. In addition, MySQL's rich community support and extensive development toolset also make the development and maintenance of serial port virtualization systems more convenient and efficient. Table 1-4 is the specific structure of the main data table.

Table 1. User Table

Serial number	Field name	Type	Size	Comment
1	ID	int	11	PK
2	username	varchar	50	
3	password	varchar	50	
4	email	varchar	50	
5	phone	varchar	20	
6	status	tinyint	1	

Table 2. Devices Table

Serial number	Field name	Type	Size	Comment
1	ID	int	11	PK
2	device_name	varchar	50	
3	device_type	varchar	50	
4	location	varchar	100	
5	phone	tinyint	1	

Table 3. Data_Transfers Table

Serial number	Field name	Type	Size	Comment
1	ID	int	11	PK
2	device_id	int	11	
3	timestamp	datetime	NULL	
4	data_size	int	11	
5	compression_ratio	float	NULL	

Table 4. System_Logs Table

Serial number	Field name	Type	Size	Comment
1	ID	int	11	PK
2	timestamp	datetime	NULL	
3	log_level	varchar	10	
4	message	varchar	255	
5	user_id	int	11	FK

4. Application of Brotli in the System

In Chapter 2.2 of this article, the advantages and disadvantages of Brotli, LZ4, Zstd, and Deflate algorithms in terms of compression rate for serial port devices are compared. The gap between Brotli and Zstd is not obvious. The following is a comparison of the compression and performance of the two algorithms under the same network conditions. The time required for decompression. Fig. 5 compares the transmission speed of data before compression and data after compression under the same network conditions. Fig. 6 compares the compression and

ISSN: 2710-0588

decompression time of the two algorithms Brotli and Zstd for the same size serial port device data in the same network environment.

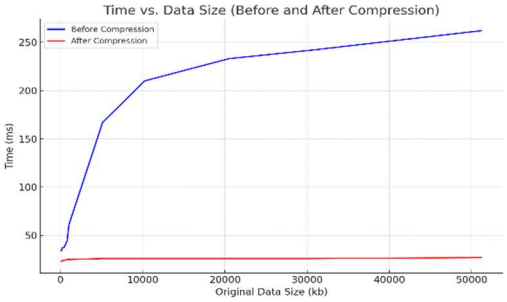

Fig. 5 Before and after compression

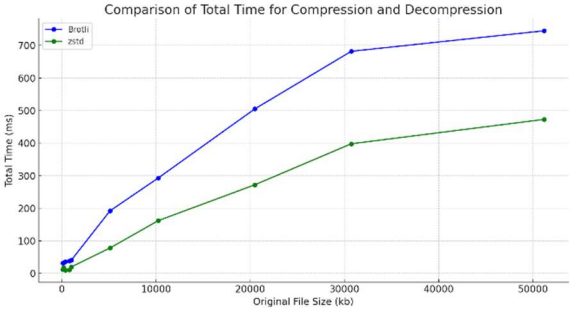

Fig. 6 Total compression and decompression time

Acknowledgments

Title: Research on classification algorithm and system of seafloor sediment based on sonar image; Liaoning Provincial Department of Education Class A Project (No. LJKZ0515).

References

[1] Xu Jie, Cai Lijun, Lu Yong, et al. Design and development of vacuum leak detection monitoring system based on configuration software and multi-serial port communication [J]. Nuclear Fusion and Plasma Physics, 2022, 42(02): 216-222 .DOI:10.16568/j.0254-6086.202202008.

[2] Qian Xinbo. Implementing Web applications to access local serial devices based on WebSocket service [J]. Computers and Networks, 2020, 46(14): 62-65.

[3] Wang Shiying, Wang Wei. Electric energy measurement automation system based on virtual serial port and configuration technology [J]. Shandong Industrial Technology, 2016(01):162-163.DOI:10.16640/j.cnki.37-1222/t.2016.01.

[4] Du Peixian. Research and application of non-networked file directory management based on compressed QR code[D]. Xi'an Petroleum University, 2024.DOI:10.27400/d.cnki.gxasc.2023. 001043.

[5] RANDY. Key technologies for edge computing load balancing based on K3S[D]. Harbin Institute of Technology, 2023.DOI:10.27061/d.cnki.ghgdu.2022.004824.

[6] Valerio HMJ. Lossless data compression, algorithm comparison and implementation[D]. Central China Normal University, 2021.DOI:10.27159/d.cnki.ghzsu.2019.002068.

[7] Thomas Pettit; Scott Cosentino, The MySQL Workshop: A practical guide to working with data and managing databases with MySQL , Packt Publishing, 2022.

[8] R. N. Aher and M. Pande, "Analysis of Lossless Data Compression Algorithm in Columnar Data Warehouse," 2022 6th International Conference On Computing, Communication, Control And Automation (ICCUBEA, Pune, India, 2022, pp. 1-4, doi: 10.1109/ICCUBEA54992.2022.10010925.

[9] F. Majerik and M. Borkovcova, "Design of Data Access Architecture Using ORM Framework," 2023 34th Conference of Open Innovations Association (FRUCT), Riga, Latvia, 2023, pp. 93-99, doi: 10.23919/FRUCT60429.2023.10328151.

[10] B. Wollmer, W. Wingerath, S. Ferrlein, F. Panse, F. Gessert and N. Ritter, "The Case for Cross-entity Delta Encoding in Web Compression (Extended)," in Journal of Web Engineering, vol. 22, no. 1, pp. 131-146, January 2023, doi: 10.13052/jwe1540-9589.2217.