Science and Technology and Social Development Proceedings Series

Article

2024 International Conference on Business Economics, Education, Arts and Social Sciences (EASS 2024)

Design of Vehicle Big Data Analysis System Based on Structured Data

Weiyan Tan [1,*]

[1] Guangdong Veterans Service Center, Guangzhou, China
* Correspondence: Weiyan Tan, Guangdong Veterans Service Center, Guangzhou, China

Abstract: In order to meet the needs of information construction of company and make full use of the increasingly large volume of traffic checkpoint data, a big data analysis system of vehicles based on structured data has been designed and implemented. The checkpoint data provided by the video structured platform was utilized by the system, which employed Kafka as the message bus, Elasticsearch as the distributed platform for storage and search, Redis as the distributed service for model and algorithm, and a warehouse for the real-time transmission, storage, retrieval, and analysis of traffic data. The system integrated multiple related data from this industry, providing more valuable clues. Following deployment and testing, the system was found to meet the business requirements in terms of functionality. In terms of performance, when the total data size reached one billion and the new data per minute reached 100,000, the operation time of the user interface (UI) response and algorithm module reached a second level, meeting the performance requirements. The system can provide the company with a range of practical application services, including accurate results, comprehensive analysis, and timely warnings.

Keywords: component; traffic checkpoint; big data; real-time; analysis

Published: 31 December 2024

1. Introduction

In recent years, China has witnessed a significant expansion in its investment in the development of public security systems. Video surveillance systems have emerged as a key area of focus in this regard. Video surveillance and tracking have emerged as the primary investigative tools employed by company at all levels. As the number of vehicle chokepoints deployed in cities continues to grow, company around the world are already in possession of a considerable quantity of vehicle chokepoint data and video image resources [1]. For instance, in Beijing, the city's daily traffic checkpoint data collection reaches a volume of ten million data points. Nevertheless, while the accumulation of video resources offers considerable assistance, it also presents significant challenges in terms of storage and analysis [2]. The challenge for company is to make full use of these resources, combined with modern information technology, for intelligence, criminal investigation, law and order, and other police services. Nevertheless, the existing vehicle-based big data analysis systems are strikingly similar, predominantly reliant on data queries, and beset by shortcomings such as a narrow range of analyzed data, protracted analysis times, and a low system utilisation rate, among others [3-5]. These systems address only some of the issues, leaving others unresolved. This system is only capable of addressing some of the issues, but it is not sufficient for the demands of in-depth analysis.

In order to enhance the utilisation of vehicle data, facilitate more comprehensive, accurate and timely analysis, and align with the evolving informational needs of company, this paper presents a vehicle big data analysis system, developed on the foundation of the completed video structured platform. The system combines a vast collection of vehicle information, business data and image data in the company network, and uses Kafka message queue, Elasticsearch distributed engine, Redis in-memory database and a large number of modeling algorithms to achieve real-time transmission, storage, retrieval and analysis of massive data. This enables the system to provide rich practical business applications.

2. System Design

2.1. System Architecture Design

The vehicle big data analysis system is based on a video structured platform, which includes a collection layer, a storage layer, a service layer and an application layer, and takes into account a security system and a standardized system. The system software architecture design is shown in Figure 1.

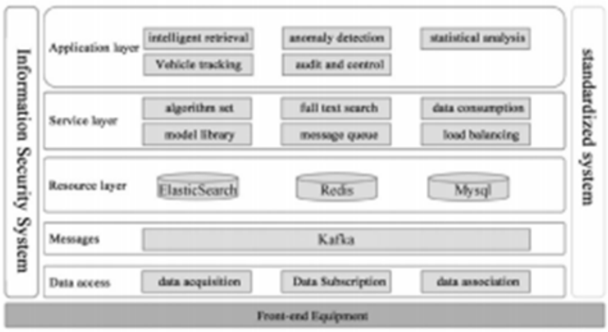

Figure 1. System Software Architecture Design View.

(1) Application layer. A Web platform developed based on Vue.js framework, providing relevant application functions for different police branches. The application functions of the platform include intelligent search, abnormal investigation, one car per file, pre-warning, statistical analysis, and 17 tactical techniques, including abnormal parking, quick departure from the city, regional wandering, concealing the face, dormancy at night and frequent night outings, frequent passing cars, first entry into the city, first appearance, key vehicle analysis, unregistered vehicles, cloned vehicles, concealed vehicles excavation, spatial collision, accompanying vehicles, landing points, similar license plates [6].

(2) Service layer. The service layer includes model algorithm service, application service, and Elasticsearch real-time full-text search service. Based on the data in the storage layer, it provides services such as algorithm analysis, model training, and intelligent search for the upper layer applications. This includes: algorithm set, model library, distributed computing center, and distributed data search engine.

(3) Data resource layer. Based on the ES (Elasticsearch) distributed search engine platform, the layer stores massive passing car data and provides support for data analysis. Redis memory database ensures the real-time retrieval of algorithm clusters. The business database uses the MySQL database to store data related to each business function.

(4) Collection layer. Using the standardized data integration interface provided by the video structure platform, a large amount of traffic data is collected from various platforms, and Kafka is used as the messaging bus for real-time data transmission.

2.2. System Logic Processing

In practice, the vehicle big data analysis system relies on the structured data provided by the video structured platform collection interface. After the data enters the card monitoring message bus, it is supplied to the ES (Elasticsearch) search engine on the one hand to satisfy the real-time retrieval of the front-end; and on the other hand, it is deposited into the Redis internal database, and is provided to the model along with the personnel thematic database data and Vehicle and Driver Management data. On the other hand, it is stored in the Redis internal database, and together with the personnel database data and Vehicle and Driver Management data, it is provided to the model algorithm module, which realizes the functions of automated labeling, one-vehicle-one-file, and alarm deployment for the front-end. The logical structure of the system is shown in Figure 2.

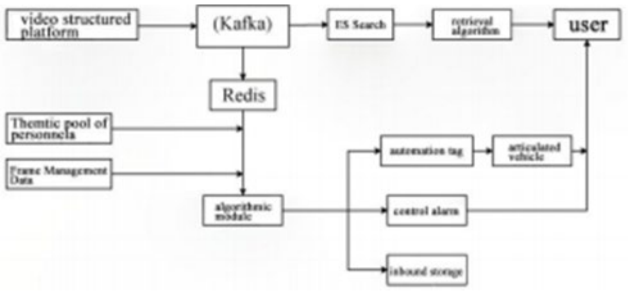

Figure 2. System Logic Processing Structure View.

2.3. System Deployment Operations

(1) System Deployment: All modules are deployed in clusters, and with the expansion of business scale and increased data access, the service nodes can also be smoothly expanded horizontally to ensure high availability of services. Under the test environment, the daily average of 4 million transit data can be processed. The system deployment is shown in Figure 3. The system deployment configuration is shown in Table 1.

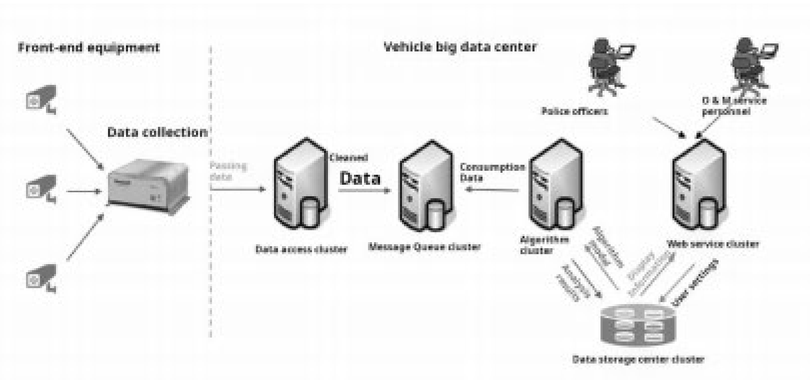

Figure 3. The implantation of information systems.

Table 1. System Deployment Configuration.

Name	Configuration	Quantity
Kafka cluster	CPU:E5-2630 V4C×2	2 sets
ES search engine	CPU:E5-2630 V4C×2	3 sets
Algorithm server	CPU:E5-2630 V4C×2	2 sets
Application Server	CPU:E5-2630 V4C×1	1 sets
Web Platform	CPU:E5-2630 V4C×1	1 sets
Middle Tier Database	CPU:E5-2630 V4C×2	1 sets

(2) Communication: The front and back ends communicate with each other based on the HTTP protocol, and the back- end data transmission is based on the Kafka message queue.

(3) Storage: The original passing data is indexed into Elasticsearch, the analysis results are stored into MySQL, and the passing data is pre-processed and cached into Redis for quick reading by the algorithm.

(4) Disaster recovery: The data storage adopts the master-slave mode and cluster deployment. Elastic-Search automatically creates a copy of the data in the search. The data in Elasticsearch will automatically create a copy of the data, even if some nodes are down, the integrity of the data can be guaranteed. The system regularly backs up the data to the data center and uses RAID5 hard disk storage to ensure data recovery.

(5) System operation: The system algorithm is driven by data collection events, using real-time message queues for data transmission, and Redis, a high-performance memory database, for data caching, and the algorithm performs distributed operations based on the cached data to ensure the quasi-real-time calculation results.

3. Key Technologies and System Implementation

3.1. Key Technologies

(1) Kafka is a high-throughput distributed publish/subscribe messaging system. Kafka can support millions of messages per second throughput even on very common hardware. In this paper, we utilize Kafka distributed message queuing technology [6] to handle massive requests and support high throughput message publishing and subscribing. The Kafka cluster is used as the message bus of the system. When the data is generated by multiple producers such as collection interfaces, subscriptions from other platforms, etc., the Kafka cluster publishes the messages to multiple consumer modules such as the ES engine and algorithms, so as to realize real-time transmission of massive data, and the structure of the Kafka message bus is illustrated in Figure 4.

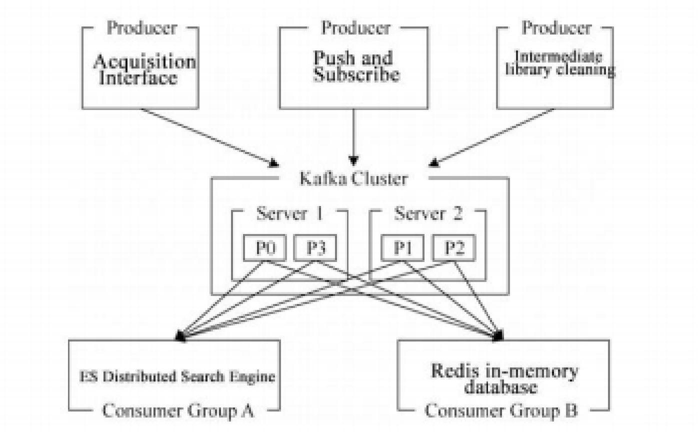

Figure 4. Kafka Message Bus Diagram.

Firstly, the user submits the data to the Elasticsearch database. Then, the corresponding statements will be segmented by the segmentation controller, and their weights and segmentation results will be stored in the data. When the user searches the data, the results are ranked and scored according to the weights, and then the results are presented to the user.

Real-time data analysis. Redis is an open source, in-memory data structure storage system. It can be used as a database, cache, and messaging middleware, and supports many types of data structures (e.g., strings, hashes, lists, sets, and ordered sets) and range queries. To ensure efficiency, Redis stores data in memory and supports certain persistence features that allow data in memory to be stored on disk and reloaded for use during restarts, which ensures the security and consistency of data in highly concurrent scenarios. Its performance is extremely high, with a read speed of 110,000 times/s and a write speed of 81,000 times/s. It also supports clustering, distributed, and master-slave synchronization configurations, and in principle, it can be infinitely scalable, allowing more data to be stored in memory. In this paper, we use Kafka message queue as a message bus to ensure real-time data transmission, and utilize the ultra-high performance of Redis to ensure real-time reading and writing of data [7] .

When new data is generated, it is released to the ES engine for storage on the one hand, and to the Redis in-memory database on the other hand, which is supplied to the model algorithm module for real-time analysis [8]. The algorithm module of the system contains more than ten kinds of business-related algorithm models: abnormal stopping, set-plate vehicle, same-traveling vehicle, stopping point, regional wandering, spatial-temporal collision, etc. The high- performance reading and writing capability of Redis allows each algorithm model to read data simultaneously, and by running various algorithms, we mark various kinds of abnormal behaviors as tags for the vehicle data, and at the same time, we store tagged data into MySQL database as the intermediate library for querying and retrieving the data. At the same time, the labeled data is stored in the MySQL database as a model intermediate library for query retrieval. The real-time analysis process is shown in Figure 5. The system analyzes every record of passing vehicles in real-time algorithms without interruption for 7 × 24 hours, generates real-time abnormal vehicle behavior data labels, and dynamically establishes an archive of every vehicle passing through the traffic checkpoints, which is used for the collision of big data analysis [9-10]. According to the vehicle abnormal behavior label and the frequency of abnormal behavior, the system will automatically carry out the warning and recommend the warning information to the Web platform. Platform users can find vehicles with abnormal behavior in the first time, so as to take measures such as deployment and control in advance, which greatly improves the ability of prior prediction, prevention and warning.

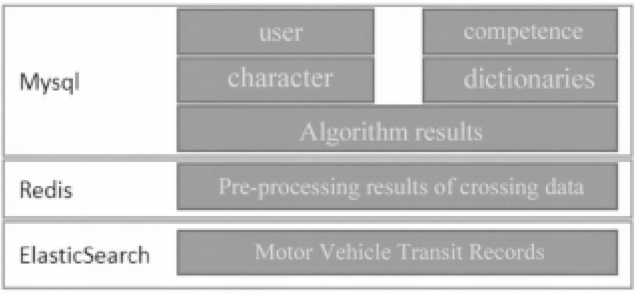

Figure 5. Database realization diagram.

Data integration and analysis. The system provides interface docking, manual entry, batch import and other forms to integrate other vehicle-related data (Vehicle and Driver's License Administration, personnel database data, stolen vehicles, etc.) into vehicle research and judgment algorithms to provide more valuable clues for vehicle research and judgment, enhance the value of the data, and improve the efficiency of case handling. By integrating the data of frame tube, the system obtains the license plate number information of the vehicle, so as to find the vehicle by the person and carry out the purpose of alarm deployment. By fusing face recognition data, hotel accommodation data, traveling data, etc., and combining with vehicle information from traffic checkpoints, the system can draw a complete human-vehicle trajectory of the control target, which can be used for in-depth big data research and analysis of the case to improve the detection rate [11-13].

3.2. System Realization

Depending on the nature of the data, three different databases, MySQL, Redis and Elasticsearch, are used for data storage and management. The database implementation is shown in Figure 5, and the comparison of the characteristics of the three databases is shown in Table 2.

Table 2. Database Features Comparison Table.

Features	Database		
	Redis	*MySQL*	*Elasticsearch*
Capacity scalability	Low	Medium	High
Query timeliness	High	Medium	High
Query flexibility	low	High	High
Read speed	Fast	Medium	fast
Coherence and transactionality	Weak	Good	Weak

The service layer is developed in java language and the system is a Windows system. The application layer is a web platform based on Vue.js framework. Figure 6 shows the main interface of the system, which contains the entry buttons of statistical analysis, intelligent search, vehicle tracking, abnormality checking and so on [14-15].

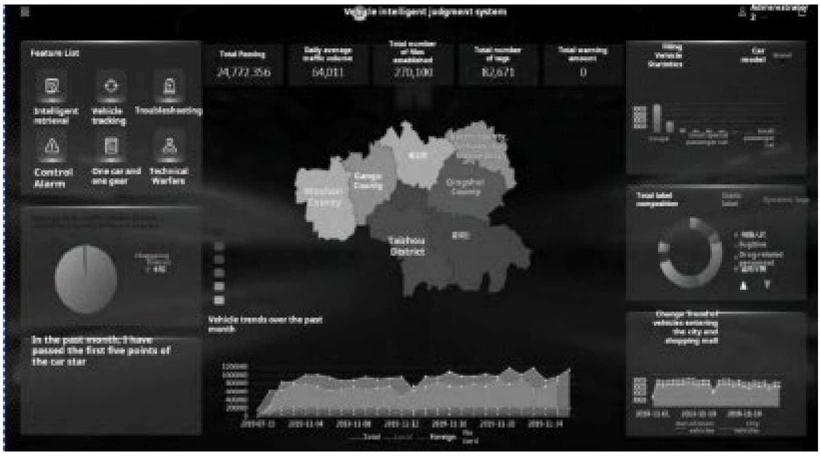

Figure 6. System Home.

4. System Testing

4.1. System Functionality Test

The system functional test mainly uses Cypress E2E (end to end) Web testing framework to test whether the system functions are realized and the operation effect is achieved by automating the UI test on the Web platform of the application layer and generating a txt file by saving the return results from each server side for business verification.

The system underwent 6 full-function tests and more than 10 random UI tests. Throughout the testing process, no critical or crashing errors were found, and the resulting files met business requirements as verified by business validation. This demonstrates that the system meets user needs in terms of functionality.

4.2. System Performance Test

System Performance Testing for Big Data. The system performance test is to monitor the response time of each module of the system while automating the UI test, and to monitor the computation time of each algorithmic model to verify the performance of each module of the system when new data is generated. Table 3 shows the average response time of each module during UI testing.

Table 3. Average response time for UI test modules.

Functionality	Module		
	ES search engine	Application server	Web platform
Intelligent retrieval	23	12	18
One Vehicle One Document	19	22	23
Statistical Analysis	246	87	25
Vehicle Tracking	32	15	29
Anomaly Detection	487	57	232
Functionality	Module		
	ES search engine	Application server	Web platform
Average	233	68	35

The test results show that the response time of each function of the system is in the order of seconds when the vehicle data reaches hundreds of millions. In the new data to reach 2,000 per second, that is, every day the new 172.8 million pieces of vehicle data. When the new data reaches 2,000 items per second, i.e., 172.8 million items of vehicle data per day (at present, except for mega cities, a large or medium-sized city has 10 million items of vehicle data per day, and the test data has exceeded the actual vehicle data generated by a city per day), the transmission, storage and calculation time of each module also reaches the second response time. The performance can almost achieve real-time analysis and real-time query, which can meet the current user needs. In the future, with the increase of vehicle ownership in the city and the continuous construction of vehicle bayonet collection points, the amount of new data will continue to increase every day, so we can consider hardware expansion by adding Kafka and ES nodes, and improve the load capacity of the system through cluster deployment to meet the real-time requirements of the system.

5. Conclusion

This article describes the design and development of a big data analysis system based on vehicle information. Compared to existing related big data systems , this system has advantages such as high timeliness, strong practicality, accurate recommendation results, and comprehensive analysis. These can be summarized in the following three aspects:

(1) Analysis methods: In contrast to existing big data systems primarily based on rule-based queries, this system has established a large number of business-related model libraries and algorithm libraries for analyzing and mining data value. (2) Timeliness: Unlike many existing big data systems that mainly perform post-event query analysis, this system can conduct real-time analysis for proactive warning and timely control. (3) Practicality: In addition to pure passing vehicle data, this article integrates personnel-specific database information for comprehensive human-vehicle integrated analysis. Currently, this system has been officially deployed in Beijing, Gansu, and other regions. It fully utilizes extensive passing vehicle data to provide more comprehensive, accurate, and intelligent application services for public companys pre-warning control measures against anomalies investigation and key vehicle management.

References

1. D. Yang, S. Zhang, T. Niu, Y. Wang, H. Xu, K. M. Zhang, and Y. Wu, "High-resolution mapping of vehicle emissions of atmospheric pollutants based on large-scale, real-world traffic datasets," *Atmos. Chem. Phys.*, vol. 19, no. 13, pp. 8831-8843, 2019, doi: 10.5194/acp-19-8831-2019.
2. M. Stiebe, "Social big data mining for the sustainable mobility and transport transition: findings from a large-scale cross-platform analysis," *Eur. Transp. Res. Rev.*, vol. 16, no. 1, 2024, doi: 10.1186/s12544-024-00651-3.
3. J. Svelch, "Normalizing player surveillance through video game infographics," *New Media Soc.*, vol. 26, no. 6, pp. 3127-3145, 2024, doi: 10.1177/14614448221097889.
4. T. A. Henzinger, R. Jhala, R. Majumdar, et al., "Abstractions from proofs," *Symp. on Principles of Programming Languages*, 2004, vol. 49, no. 4, pp. 232-244, doi: 10.1145/964001.964021.
5. T. C. K. Andersen, A. Aagaard, and M. Magnusson, "Exploring business model innovation in SMEs in a digital context: Organizing search behaviours, experimentation and decision-making," *Creat. Innov. Manag.*, vol. 31, no. 1, pp. 19–34, 2022., doi: 10.1111/caim.12474.
6. Q. Y. Jiang et al., "Research on teaching optimization of internet plus classroom enabled by big data technology," in *Proc. 2023 2nd Int. Conf. Sport Sci., Educ. Social Develop. (SSESD 2023)*, vol. 781. Springer Nature, 2023. doi: 10.2991/978-2-38476-122-7_4.
7. D. Bounov, A. DeRossi, M. Menarini, W. G. Griswold, and S. Lerner, "Inferring loop invariants through gamification," in *Proc. 2018 CHI Conf. Human Factors Comput. Syst.*, Apr. 2018, pp. 1–13, doi: 10.1145/3173574.3173805.
8. D. Yang, S. Zhang, T. Niu, Y. Wang, H. Xu, K. M. Zhang, and Y. Wu, "High-resolution mapping of vehicle emissions of atmospheric pollutants based on large-scale, real-world traffic datasets," *Atmos. Chem. Phys.*, vol. 19, no. 13, pp. 8831–8843, 2019. doi: 10.5194/acp-19-8831-2019.
9. Q. L. Le, J. Li, J. Sun, et al., "Automatic loop-invariant generation and refinement through selective sampling," in *IEEE/ACM Int. Conf. on Automated Software Engineering*, IEEE Computer Society, 2017, doi: 10.1109/ASE.2017.8115689.
10. Q. Long, "Investigation on big data evaluation and visualization of Internet of Things based on edge computing," *Meas. Sens.*, 2024, doi: 10.1016/j.measen.2024.101177.
11. M. A. Saleem, X. Li, K. Mahmood, T. Tariq, M. J. Alenazi, and A. K. Das, "Secure RFID-assisted authentication protocol for vehicular cloud computing environment," *IEEE Trans. Intell. Transp. Syst.*, 2024, doi: 10.1109/TITS.2024.3371464.
12. P. Deshpande and A. Rasin, "Correlation Aware Relevance-Based Semantic Index for Clinical Big Data Repository," *J. Imaging Informatics Med.*, 2024, doi: 10.1007/s10278-024-01095-w.
13. G. Wunsch, "Big data analytics: A demographer's perspective," *Bull. Sociol. Methodol./Bull. Méthodologie Sociologique*, vol. 162, no. 1, pp. 243-255, 2024, doi: 10.1177/07591063241236071.
14. M. H. Touny, S. A. Moussa, and S. A. Hadi, "Scalable fuzzy multivariate outliers identification towards big data applications," *Appl. Soft Comput.*, pp. 15-18, 2024, doi: 10.1016/j.asoc.2024.111444.
15. Tran, H. N, "Vietnamese enterprises' considerations of big data and analytics implementation post-COVID-19 pandemic," *Int. J. Organ. Anal.*, vol. 32, no. 1, pp. 95-107, 2024, doi: 10.1108/IJOA-12-2022-3545.